



**Arun Jagatheesan**

**Reagan Moore**

**San Diego Supercomputer Center (SDSC)**

**University of California, San Diego**

**{arun, moore} @sdsc.edu**

**NASA/IEEE MSST 2004**

**12th NASA Goddard/21st IEEE Conference on  
Mass Storage Systems & Technologies**

**The Inn and Conference Center  
University of Maryland University College  
Adelphi MD USA**

**April 13-16, 2004**





# Storage Resource Broker

- **Distributed data management technology**
  - Developed at San Diego Supercomputer Center (Univ. of California, San Diego)
  - 1996 - DARPA Massive Data Analysis
  - 1998 - DARPA/USPTO Distributed Object Computation Testbed
  - 2000 to present - NSF, NASA, NARA, DOE, DOD, NIH, NLM, NHPRC
- **Applications**
  - Data grids - data sharing
  - Digital libraries - data publication
  - Persistent archives - data preservation
  - Used in national and international projects in support of Astronomy, Bio-Informatics, Biology, Earth Systems Science, Ecology, Education, Geology, Government records, High Energy Physics, Seismology

# Acknowledgement: SDSC SRB Team



- Arun Jagatheesan
- George Kremenek
- Sheau-Yen Chen
- Arcot Rajasekar
- Reagan Moore
- Michael Wan
- Roman Olschanowsky
- Bing Zhu
- Charlie Cowart

## *Not In Picture:*

- Wayne Schroeder
- Tim Warnock(BIRN)
- Lucas Gilbert
- Marcio Faerman (SCEC)
- Antoine De Torcy

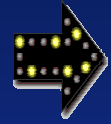
## **Students:**

Xi (Cynthia) Sheng  
Allen Ding  
Grace Lin  
Jonathan Weinberg  
Yufang Hu  
Yi Li

## **Emeritus:**

Vicky Rowley (BIRN)  
Qiao Xin  
Daniel Moore  
Ethan Chen  
Reena Mathew  
Erik Vandekieft  
Ullas Kapadia

# Tutorial Outline



## Introduction

- Data Grids
- Data Grid Infrastructures
- **Information Management using Data Grids**
  - Data Grid Transparencies and concepts
  - Peer-to-peer Federation of Data Grids
- **Gridflows and Data Grids**
  - Need for Gridflows
  - Data Grid Language and SDSC Matrix Project
- **Lets build a Data Grid**
  - Using SDSC SRB Data Grid Management System and its Interfaces

# Data Grids

- **Coordinated Cyberinfrastructure**
  - Formed by coordination of multiple autonomous organizations
  - Preserves local autonomy and provides global consistency
- **Logical Namespaces (Virtualizations)**
  - Virtualization mechanisms for *resources* (including storage space, data, metadata, processing pipelines and inter-organizational users)
  - Location and infrastructure independent logical namespace with persistent identifiers for all resources

# Data Grid Goals

- **Automate all aspects of data analysis**
  - Data discovery
  - Data access
  - Data transport
  - Data manipulation
- **Automate all aspects of data collections**
  - Metadata generation
  - Metadata organization
  - Metadata management
  - Preservation

# Using a Data Grid – *in Abstract*

*Data Grid*

Ask for data

Data delivered

- User asks for data from the data grid
- The data is found and returned
- Where & how details are managed by data grid
- But access controls are specified by owner





# Tutorial Outline

- **Introduction**

- Data Grids
- Data Grid Infrastructures



- **Information Management using Data Grids**

- Data Grid Transparencies and concepts
- Peer-to-peer Federation of Data Grids

- **Gridflows and Data Grids**

- Need for Gridflows
- Data Grid Language and SDSC Matrix Project

- **Data Grids and You**

- Open Research Issues and Global Grid Forum Community

- **Lets build a Data Grid**

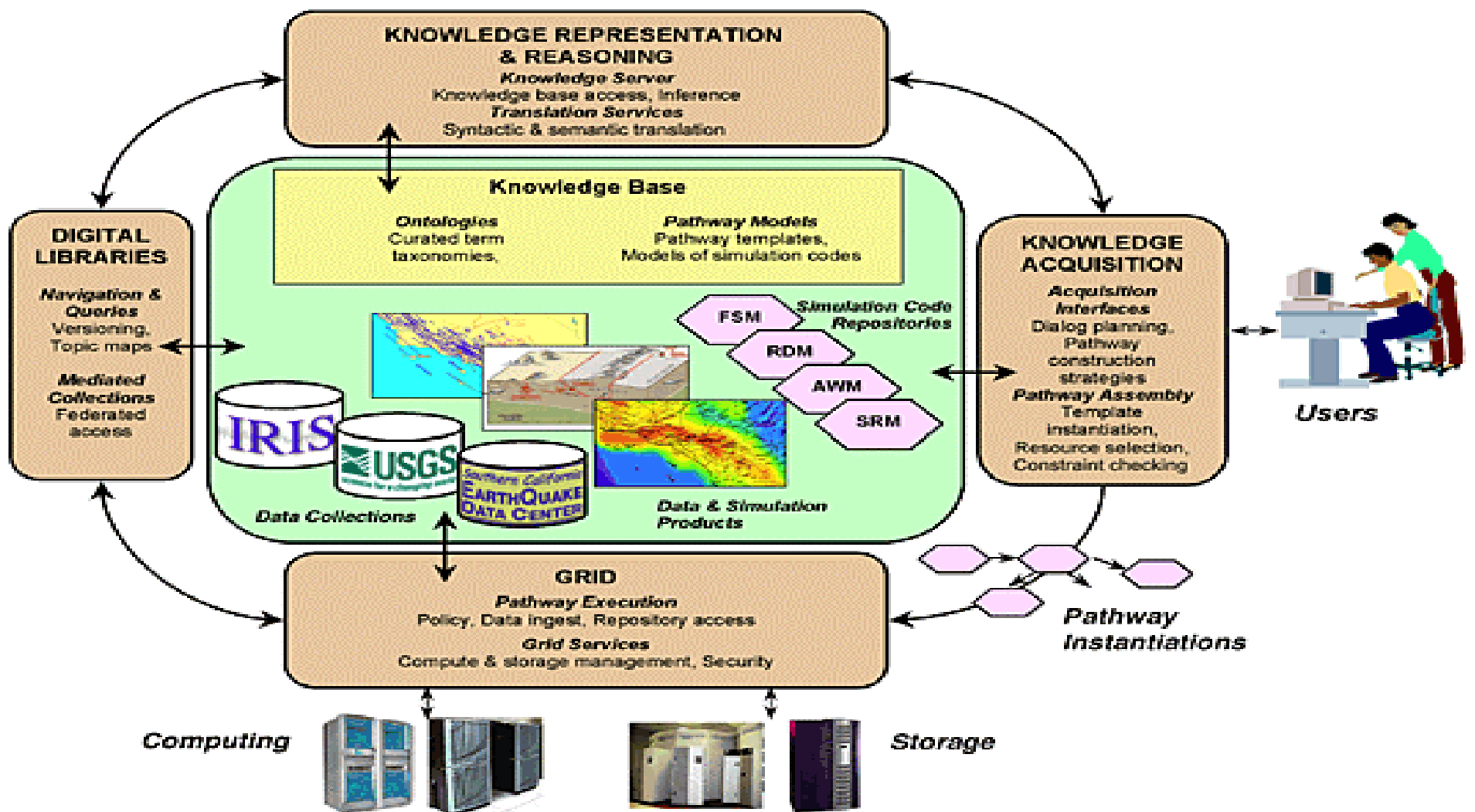
- Using SDSC SRB Data Grid Management System and its Interfaces



# SRB Environments

- **NSF Southern California Earthquake Center digital library**
- **Worldwide Universities Network data grid**
- **NASA Information Power Grid**
- **NASA Goddard Data Management System data grid**
- **DOE BaBar High Energy Physics data grid**
- **NSF National Virtual Observatory data grid**
- **NSF ROADnet real-time sensor collection data grid**
- **NIH Biomedical Informatics Research Network data grid**
- **NARA research prototype persistent archive**
- **NSF National Science Digital Library persistent archive**
- **NHPRC Persistent Archive Testbed**

# Southern California Earthquake Center



# Southern California Earthquake Center

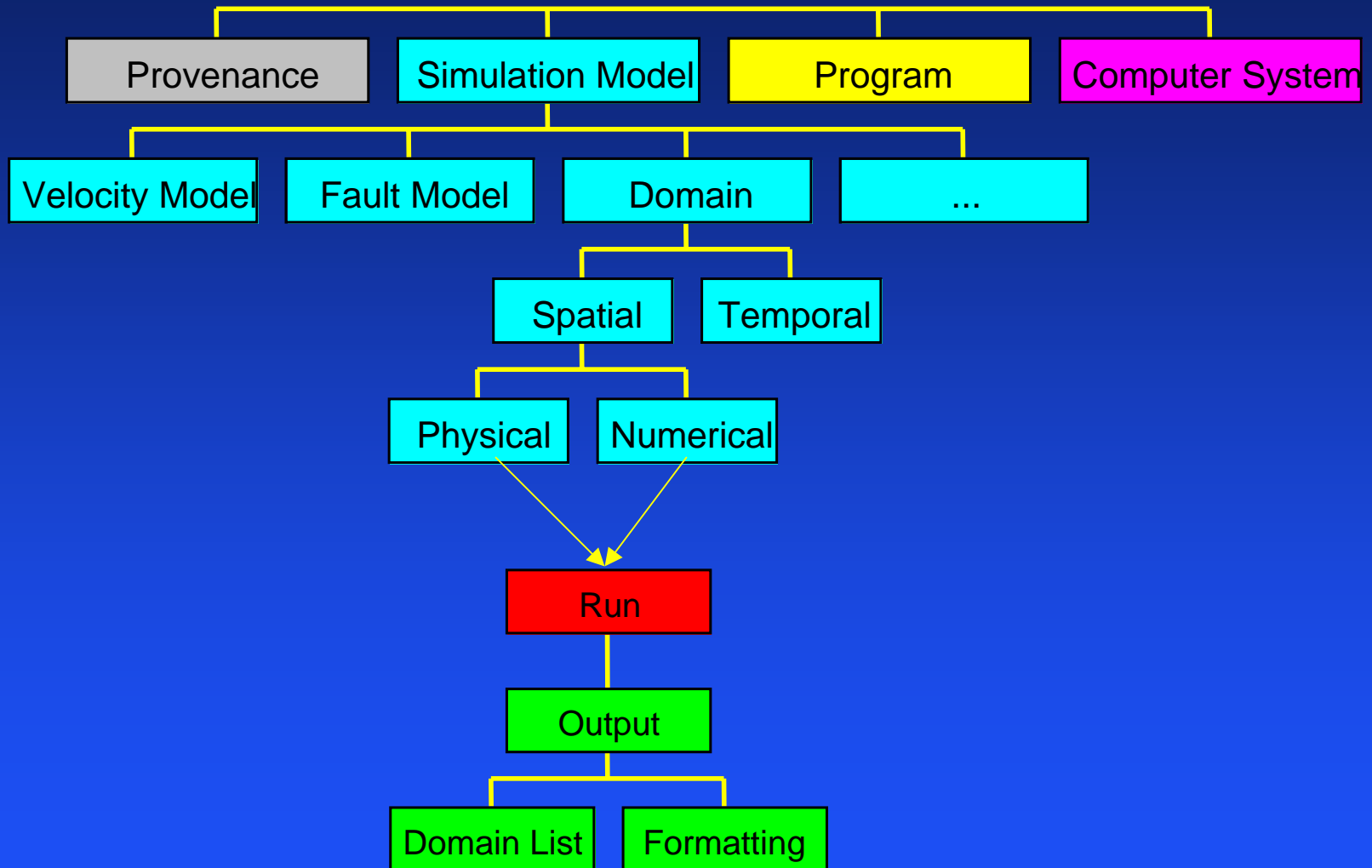
- **Build community digital library**
- **Manage simulation and observational data**
  - Anelastic wave propagation output
  - 10 TBs, 1.5 million files
- **Provide web-based interface**
  - Support standard services on digital library
- **Manage data distributed across multiple sites**
  - USC, SDSC, UCSB, SDSU, SIO
- **Provide standard metadata**
  - Community based descriptive metadata
  - Administrative metadata
  - Application specific metadata

# SCEC Digital Library Technologies

- **Portals**
  - Knowledge interface to the library, presenting a coherent view of the services
- **Knowledge Management Systems**
  - Organize relationships between SCEC concepts and semantic labels
- **Process management systems**
  - Data processing pipelines to create derived data products
- **Web services**
  - Uniform capabilities provided across SCEC collections
- **Data grid**
  - Management of collections of distributed data
- **Computational grid**
  - Access to distributed compute resources
- **Persistent archive**
  - Management of technology evolution

# Metadata Organization

## (Domain View versus Run View)





Explore SRB

- container
- g1
- home
  - gely.ucsd
    - foam\_small
    - mfaerman2.sdsc
      - SCEC-CME
        - Simulations
          - DFM
            - CHECKPOINT
            - Input
            - Output
              - Stress
              - Velocity
                - 00001
                - 00002
                - 00003
                - 00004
                - 00005
                - 00006
                - 00007
                - 00008
                - 00009
                - 00010
                - 00011
                - 00012
                - 00013
                - 00014
                - 00015
                - 00016
                - 00017
                - 00018
                - 00019
                - 00020

View All Metadata

Collection: 00003  
Parent Collection: /home/mfaerman2.sdsc/SCEC-CME/Simulations/DFM/Output/Velocity  
Owner: mfaerman2@sdsc

Metadata describing entity 00003

Field Name : Value Units < Explanation >

TimeStep : 3

DC.description : 4D wave propagation data

DC.subject.scheme : LCC

DC.subject.content : QE521-545

DC.type.scheme : DCMIType

DC.type.content : Dataset

DC.rights.Lang : en

DC.date.created : Thu Aug 28 09:09:11 PDT 2003

DC.creator.Name : Geoffrey Ely

DC.creator.Email : gely@ucsd.edu

DC.contributor : Qiao Xin, Marcio Faerman

DC.publisher.Name : Marcio Faerman

DC.publisher.Email : mfaerman@sdsc.edu

ProgramInfo.Name : dfm

ProgramInfo.Description : DFM: Dynamic Fault Model

ProgramInfo.Version.User : Thu Aug 28 09:09:11 PDT 2003

ProgramInfo.Authors : Geoffrey Ely, Steve Day, Boris Shkoller

ProgramInfo.Affiliation : UCSD, SDSU

ProgramInfo.ProgrammingLanguages : C, Fortran



BackForwardReloadStop

https://srb.npaci.edu/cgi-bin/new/mysrb2.cgi

SearchPrint

MailAIMHomeNetscapeSearchBookmarks

MYSRB V8.0

Explore SRB

container

g1

home

gely.ucsd

foam\_small

mfaerman2.sdsc

SCEC-CME

Simulations

DFM

CHECKPOINT

Input

Output

Stress

Velocity

00001

00002

00003

00004

00005

00006

00007

00008

00009

00010

00011

00012

00013

00014

00015

00016

00017

00018

00019

00020

00021

00022

00023

00024

00025

ProgramInfo.SystemSoftware : **MPICH, bash, gawk, gmake, perl**

ProgramInfo.Version.User : **Thu Aug 28 09:09:11 PDT 2003**

PhysicalInputs.CoordinateSystem : **Cartesian XYZ**

PhysicalInputs.Volume.MinX : **0**

PhysicalInputs.Volume.MaxX : **2.807**

PhysicalInputs.Volume.MinY : **0**

PhysicalInputs.Volume.MaxY : **5.607**

PhysicalInputs.Volume.MinZ : **0**

PhysicalInputs.Volume.MaxZ : **5.607**

PhysicalInputs.TimeSeries.EndTime : **0.15**

NumericalInputs.TimeSeries.NumberOfSteps : **3000**

NumericalInputs.TimeSeries.TimeIncrement : **5e-05**

NumericalInputs.3DData.NumPointsXdirection : **401**

NumericalInputs.3DData.NumPointsYdirection : **801**

NumericalInputs.3DData.NumPointsZdirection : **801**

NumericalInputs.3DData.GridIncrement.Xdirection : **0.007**

NumericalInputs.3DData.GridIncrement.Ydirection : **0.007**

NumericalInputs.3DData.GridIncrement.Zdirection : **0.007**

NumericalInputs.FreeSurfaceBoundaryCondition : **natural**

RunID.Description : **High resolution DFM foam rubber simulation**

RunID.Submitter.Name : **Qiao Xin**

RunID.Investigator.Name : **Geoffrey Ely**

RunID.Date.Submitted : **Thu Aug 28 09:09:11 PDT 2003**

SystemID.HostName : **tf005i**

SystemID.OS.Name : **AIX**

SystemID.OS.Version : **1**

SystemID.OS.Description : **AIX tf005i 1 5 006025764C00, SDSC Blue Horizon**

SystemID.NumberOfProcessors : **512**

OutputDescription.OutputFormat : **4 byte floats**

OutputDescription.StorageFormat : **4 byte floats**

OutputDescription.Endian : **big**





# NASA Data Grids

- **NASA Information Power Grid**
  - NASA Ames, NASA Goddard
  - Distributed data collection using the SRB
- **ESIP federation**
  - Led by Joseph JaJa (U Md)
  - Federation of ESIP data resources using the SRB
- **NASA Goddard Data Management System**
  - Storage repository virtualization (Unix file system, Unitree archive, DMF archive) using the SRB
- **NASA EOS Petabyte store**
  - Storage repository virtualization for EMC persistent store using the Nirvana version of SRB



# Data Assimilation Office

**HSI** has implemented metadata schema in SRB/MCAT

**Origin:** host, path, owner, uid, gid, perm\_mask, [times]

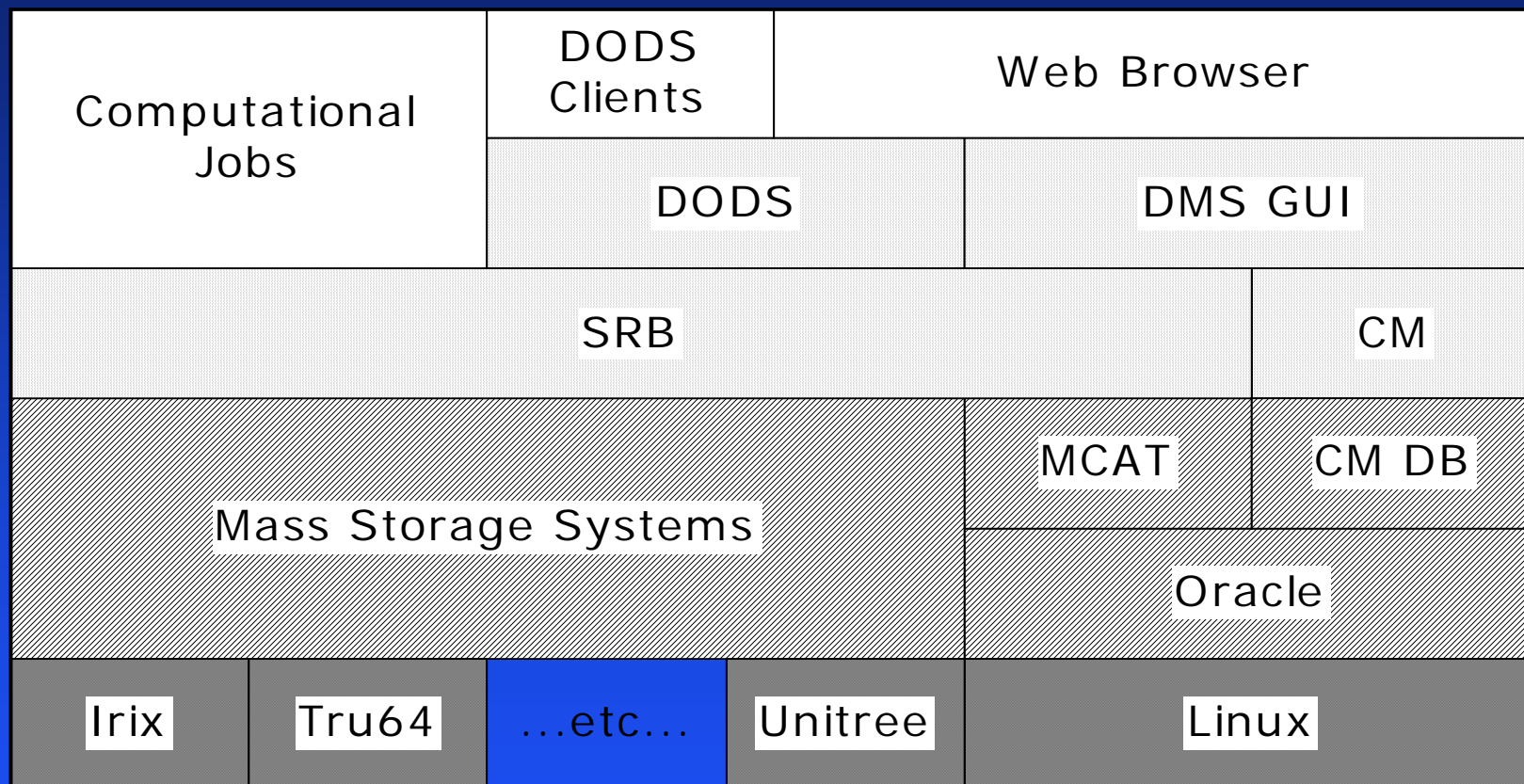
**Ingestion:** date, user, user\_email, comment

**Generation:** creator (name, uid, user, gid), host (name, arch, OS name & flags), compiler (name, version, flags), library, code (name, version), accounting data

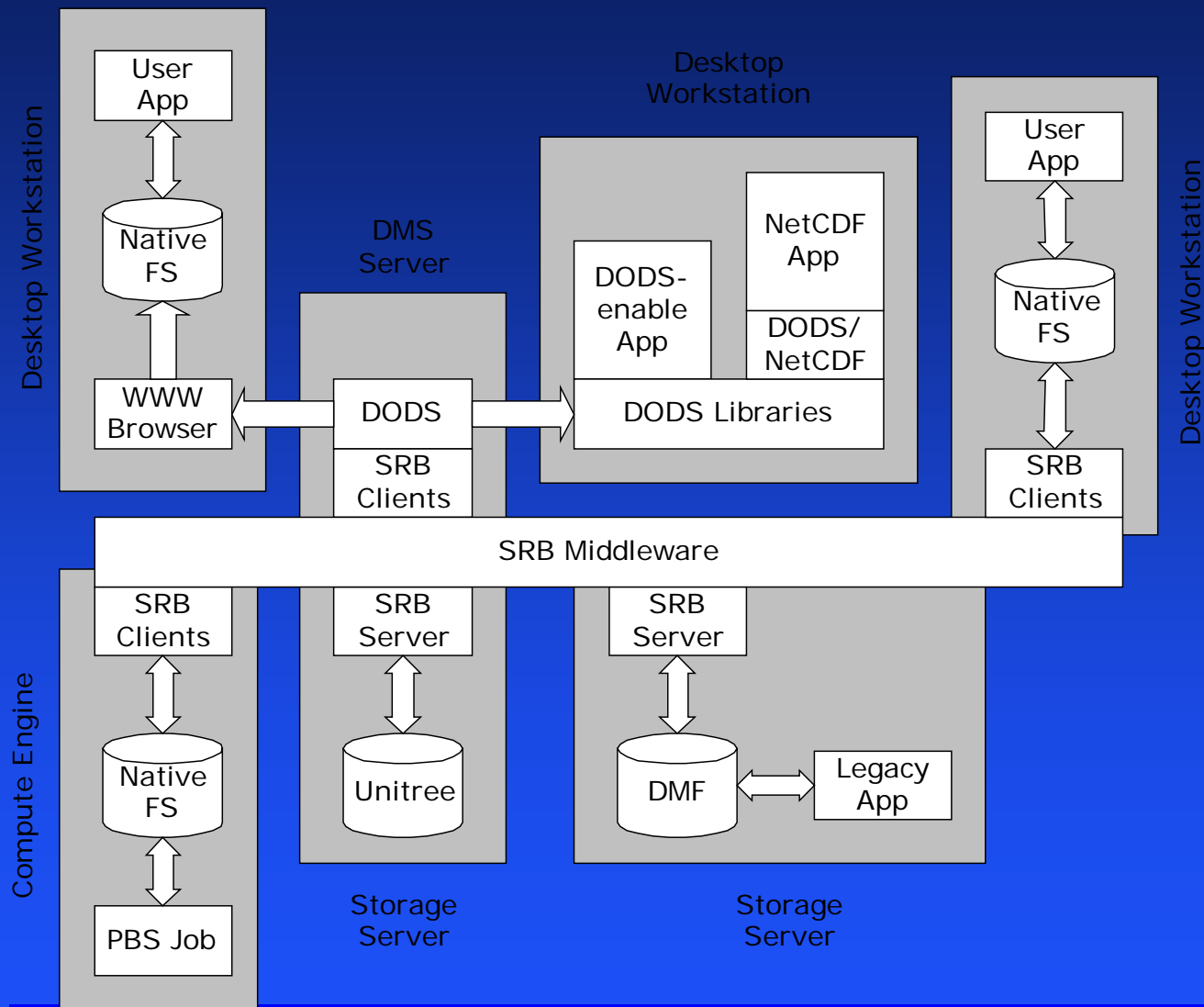
**Data description:** title, version, discipline, project, language, measurements, keywords, sensor, source, prod. status, temporal/spatial coverage, location, resolution, quality

**Fully compatible with GCMD**

# Data Management System: Software Architecture



# DODS Access Environment Integration



# National Virtual Observatory Data Grid

## 1. Portals and Workbenches

2. Knowledge  
& Resource  
Management

3. Metadata View   Data View   Catalog Analysis   Bulk Data Analysis

Concept space

Standard APIs and Protocols

4. Grid Security  
Caching  
Replication  
Backup  
Scheduling

5. Information Discovery   Metadata delivery   Data Discovery   Data Delivery

Standard Metadata format, Data model, Wire format

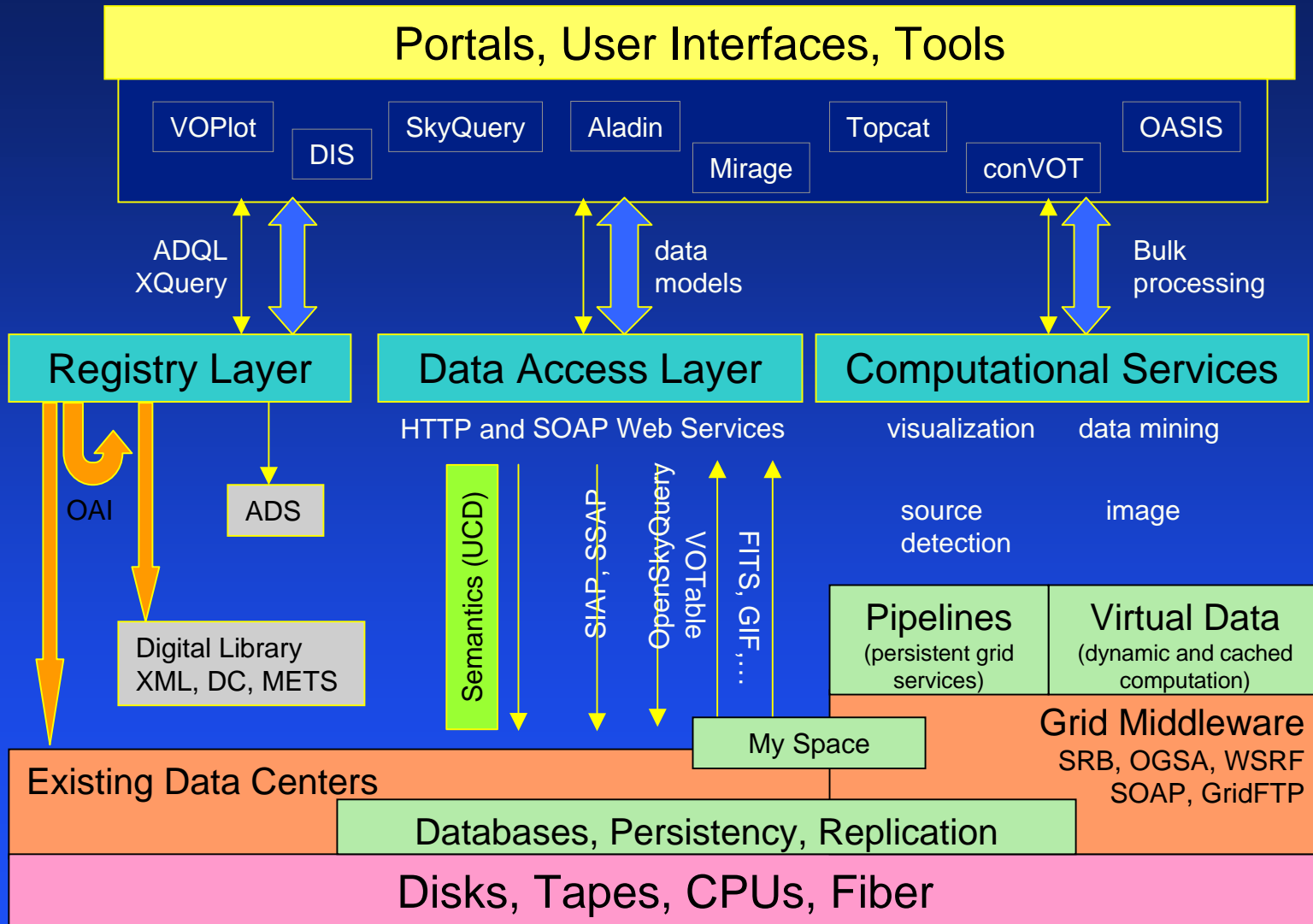
6. Catalog Mediator   Data mediator

Catalog/Image   Specific Access

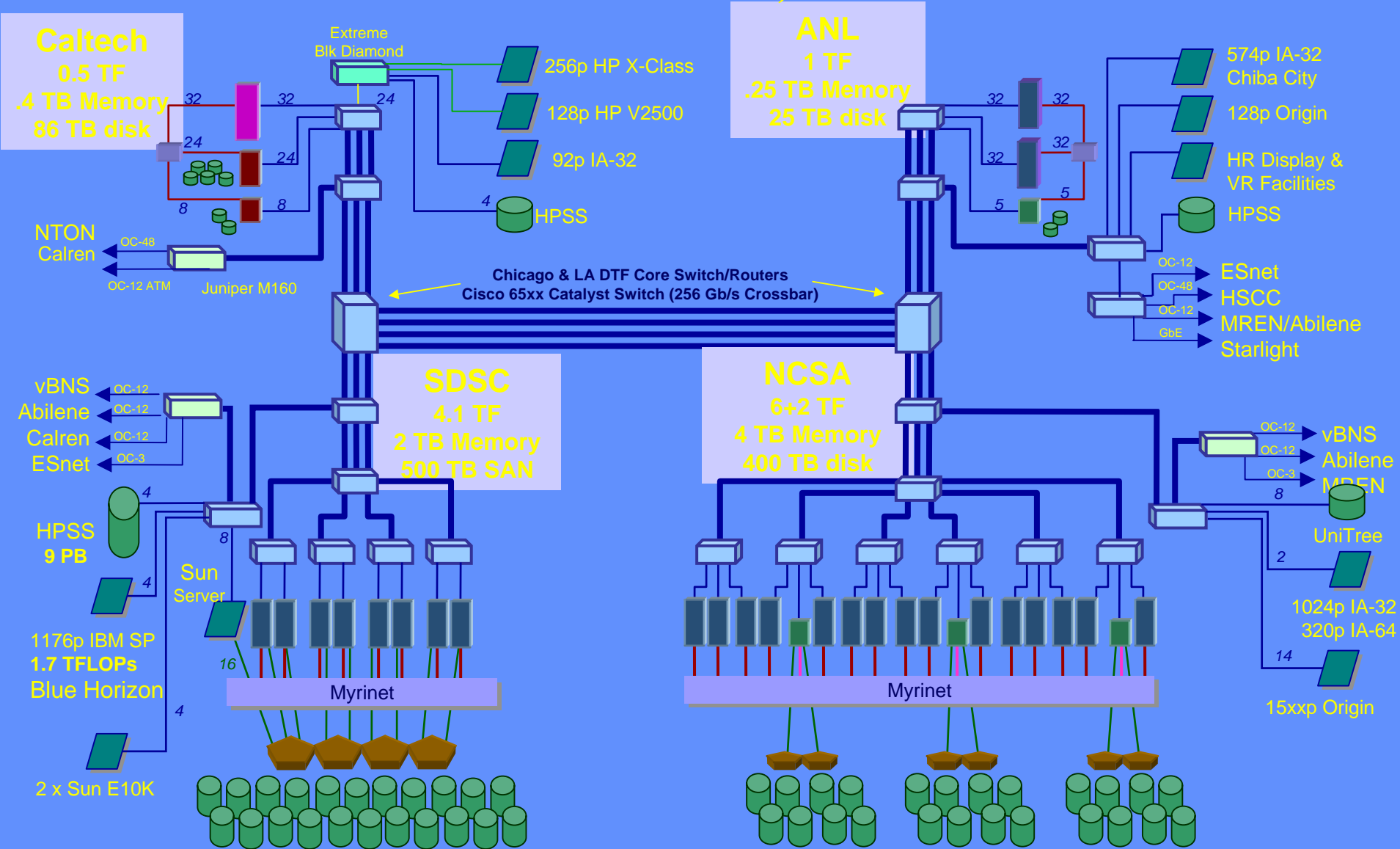
7. Compute Resources   Derived Collections   Catalogs   Data Archives



# National Virtual Observatory



# TeraGrid: 13.6 TF, 6.8 TB memory, 900 TB network disk, 10 PB archive

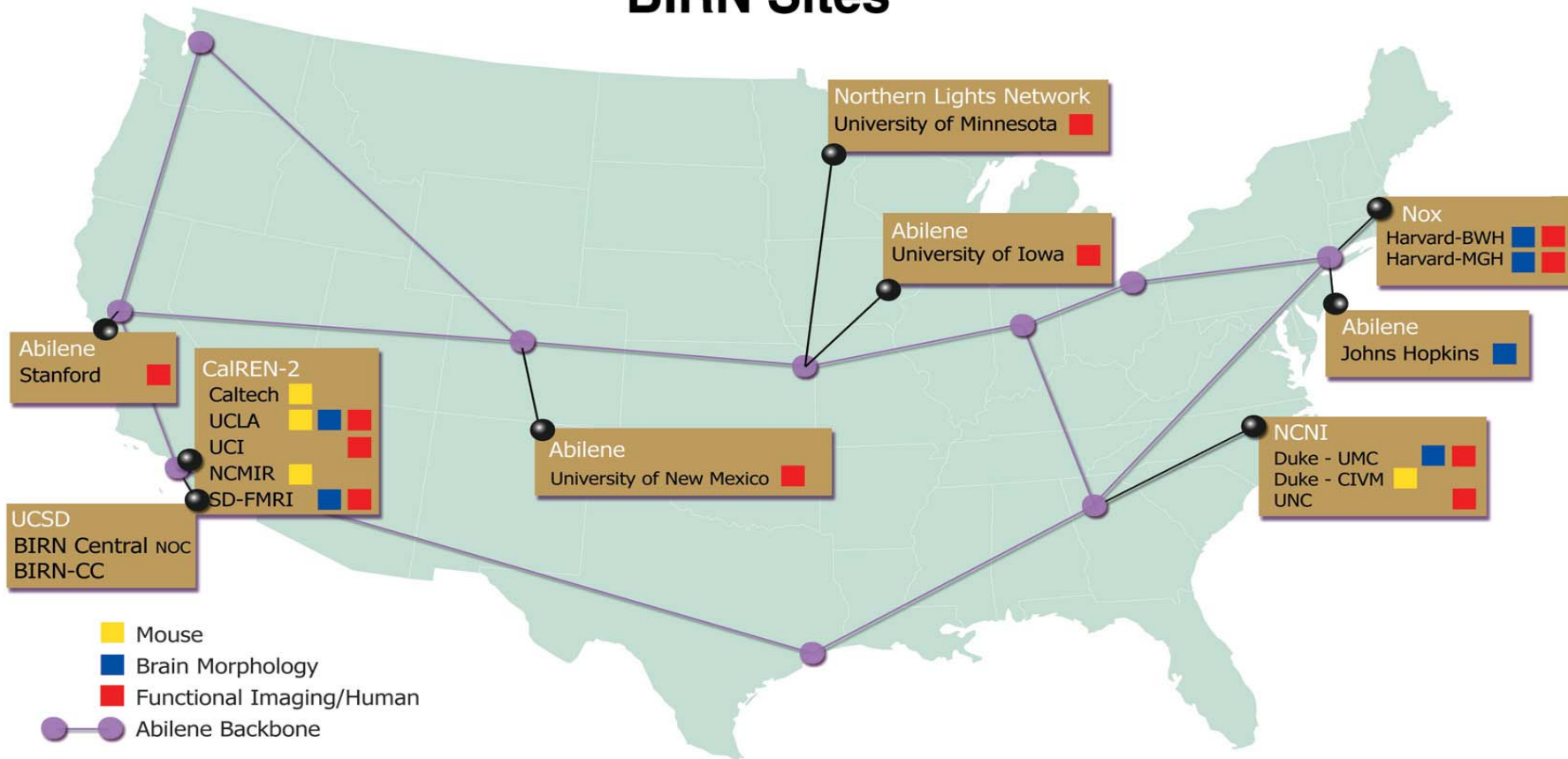




# NIH BIRN SRB Data Grid



## BIRN Sites



# SRB Collections at SDSC

	As of 12/22/2000		As of 5/17/2002		As of 3/3/2004		
<i>Project Instance</i>	<i>Data_size (in GB)</i>	<i>Count (files)</i>	<i>Data_size (in GB)</i>	<i>Count (files)</i>	<i>Data_size (in GB)</i>	<i>Count (files)</i>	<i>Users</i>
<b>Data Grid</b>							
Digsky	7,599.00	3,630,300	17,800.00	5,139,249	45,939.00	8,685,572	80
NPACI	329.63	46,844	1,972.00	1,083,230	13,700.00	4,050,863	379
Hayden			6,800.00	41,391	7,835.00	60,001	168
SLAC			514.00	77,168	3,432.00	446,613	43
LDAS/SALK			239.00	1,766	2,002.00	14,427	66
TeraGrid					22,563.00	452,868	2,585
BIRN					892.00	2,472,299	160
<b>Digital Library</b>							
DigEmbryo	124.30	2,479	433.00	31,629	720.00	45,365	23
HyperLter	28.94	69	158.00	3,596	215.00	5,110	29
Portal			33.00	5,485	1,610.00	46,278	374
AfCS			27.00	4,007	236.00	42,987	21
NSDL/SIO Exp			19.20	383	1,217.00	193,888	26
TRA			5.80	92	92.00	2,387	26
SCEC					12,311.00	1,730,432	47
UCSDLib					127.00	202,445	29
<b>Persistent Archive</b>							
NARA/Collection			7.00	2,455	72.00	82,192	58
NSDL/CI					1,529.00	12,658,072	116
<b>TOTAL</b>	<b>8 TB</b>	<b>3.7 million</b>	<b>28 TB</b>	<b>6.4 million</b>	<b>114 TB</b>	<b>31 million</b>	<b>4230</b>

\*\* Does not cover data brokered by SRB spaces administered outside SDSC.

Does not cover databases; covers only files stored in file systems and archival storage systems

Does not cover shadow-linked directories

# Commonality in all these projects

- **Distributed data management**
  - Data Grids, Digital Libraries, Persistent Archives,
  - Workflow/dataflow Pipelines, Knowledge Generation
- **Data sharing across administrative domains**
  - Common name space for all registered digital entities
- **Data publication**
  - Browsing and discovery of data in collections
- **Data Preservation**
  - Management of technology evolution

# Common Data Grid Components

- **Federated client-server architecture**
  - Servers can talk to each other independently of the client
- **Infrastructure independent naming**
  - Logical names for users, resources, files, applications
- **Collective ownership of data**
  - Collection-owned data, with infrastructure independent access control lists
- **Context management**
  - Record state information in a metadata catalog from data grid services such as replication
- **Abstractions for dealing with heterogeneity**

# Tutorial Outline

- **Introduction**

- Data Grids
- Data Grid Infrastructures



## **Information Management using Data Grids**

- Data Grid Transparencies and concepts
- Peer-to-peer Federation of Data Grids

- **Gridflows and Data Grids**

- Need for Gridflows
- Data Grid Language and SDSC Matrix Project

- **Lets build a Data Grid**

- Using SDSC SRB Data Grid Management System and its Interfaces

# Information Management Technologies

- **Data collecting**
  - **Sensor systems**, object ring buffers and portals
- **Data organization**
  - **Collections**, manage data context
- **Data sharing**
  - **Data grids**, manage heterogeneity
- **Data publication**
  - **Digital libraries**, support discovery
- **Data preservation**
  - **Persistent archives**, manage technology evolution
- **Data analysis**
  - **Processing pipelines**, manage knowledge extraction

# Assertion

- **Data Grids provide the underlying abstractions required to support all information technologies**
  - Collection building
    - Metadata extraction
  - Digital libraries
    - Curation processes
    - Distributed collections
    - Discovery and presentation services
  - Persistent archives
    - Management of technology evolution
    - Preservation of authenticity



# Information Management Terms

- **Data**
  - Bits - zeros and ones
- **Digital Entity**
  - The bits that form an image of reality (file, object, image, data, metadata, string of bits, structured sets of string of bits)
- **Metadata**
  - Semantic labels and the associated data
- **Information**
  - Semantic labels applied to data and its semantic properties
- **Knowledge**
  - Relationships between semantic labels associated with the data
  - Relationships used to assert the application of a semantic label

# Information Management data types

- **Collection**
  - The organization of digital entities to simplify management and access.
- **Context**
  - The information that describes the digital entities in a collection.
- **Content**
  - The digital entities in a collection

# Types of Context Metadata

- **Descriptive**
  - Provenance information, discovery attributes
- **Administrative**
  - Location, ownership, size, time stamps
- **Structural**
  - Data model, internal components
- **Behavioral**
  - Display and manipulation operations
- **Authenticity**
  - Audit trails, checksums, access controls

# Some Metadata Standards

- **METS - Metadata Encoding Transmission Standard**
  - Defines standard structure and schema extension
- **OAIS - Open Archival Information System**
  - Preservation packages for submission, archiving, distribution
- **OAI - Open Archives Initiative**
  - Metadata retrieval based on Dublin Core provenance attributes

# Data Management Mechanisms

- **Curation**
  - The process of creating the context
- **Closure**
  - Assertion that the collection has global properties, including completeness and homogeneity under specified operations
- **Consistency**
  - Assertion that the context represents the content

# Storage Resource Broker

- **Implements data management mechanisms needed to automate**
  - Collection building
  - Context management
  - Content management
  - Curation processes
  - Closure and validation processes
  - Consistency guarantees
- **Provides virtualization mechanisms to manage**
  - Distribution across administrative domains
  - Heterogeneous storage resources

# Data Grid Transparencies/Virtualizations (bits,data,information,..)

Semantic data Organization (with behavior)

myActiveNeuroCollection

patientRecordsCollection

Inter-  
organizational  
Information  
Storage  
Management

Virtual Data Transparency

image.cgi

image.wsdl

image.sql

Data Replica Transparency

image\_0.jpg...image\_100.jpg

Data Identifier Transparency

E:\srbVault\image.jpg /users/srbVault/image.jpg Select ... from srb.mdas.td where...

Storage Location Transparency



Storage Resource Transparency





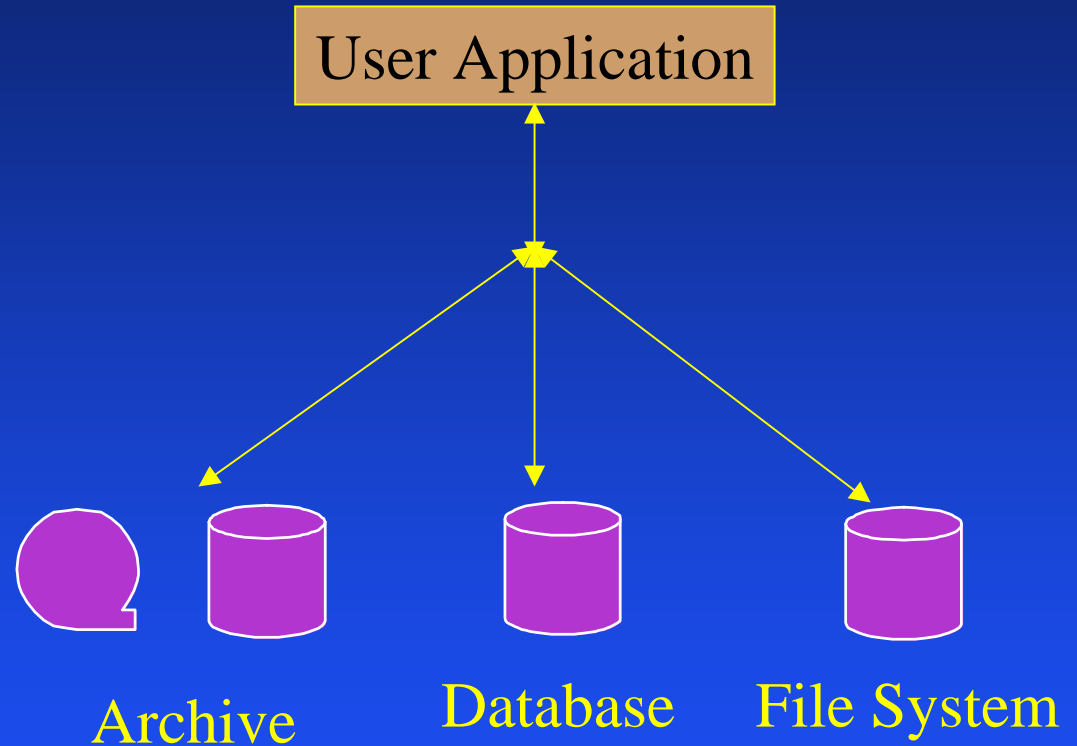
# Data Grid Transparencies

- **Find data without knowing the identifier**
  - Descriptive attributes
- **Access data without knowing the location**
  - Logical name space
- **Access data without knowing the type of storage**
  - Storage repository abstraction
- **Retrieve data using your preferred API**
  - Access abstraction
- **Provide transformations for any data collection**
  - Data behavior abstraction

# Data Grid Abstractions

- **Storage repository virtualization**
  - Standard operations supported on storage systems
- **Data virtualization**
  - Logical name space for files - Global persistent identifier
- **Information repository virtualization**
  - Standard operations to manage collections in databases
- **Access virtualization**
  - Standard interface to support alternate APIs
- **Latency management mechanisms**
  - Aggregation, parallel I/O, replication, caching
- **Security interoperability**
  - GSSAPI, inter-realm authentication, collection-based authorization

# Storage Repository Virtualization



# Storage Repository Virtualization

Remote operations

Unix file system

Latency management

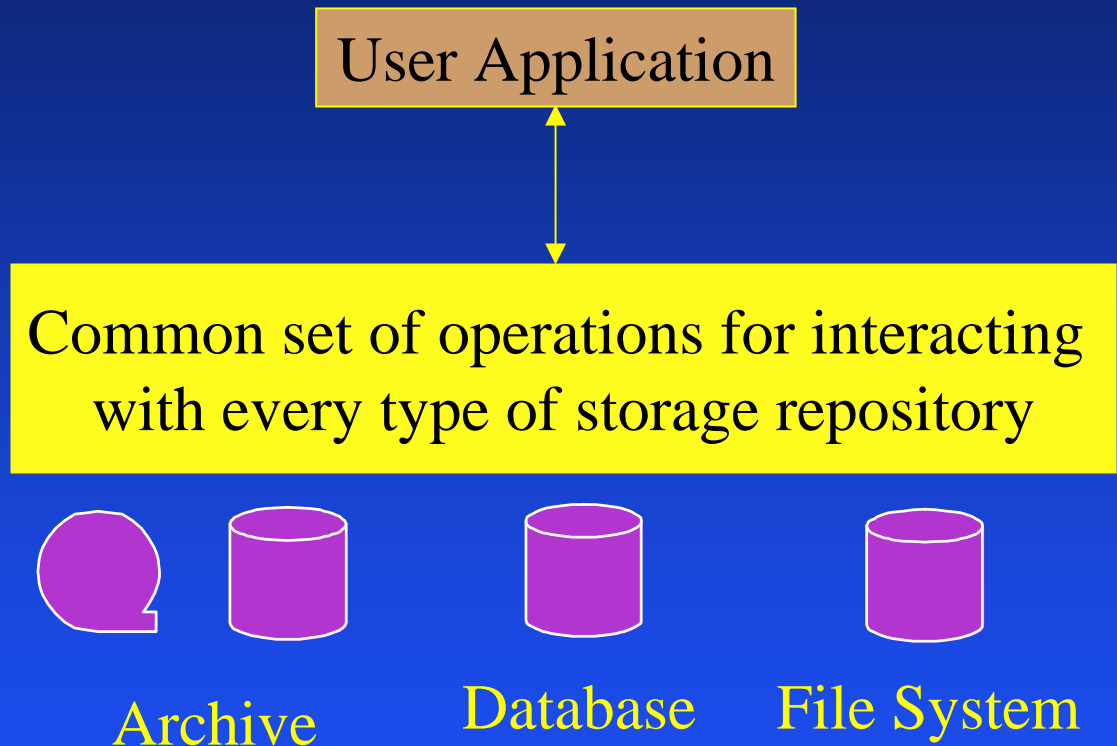
Procedures

Transformations

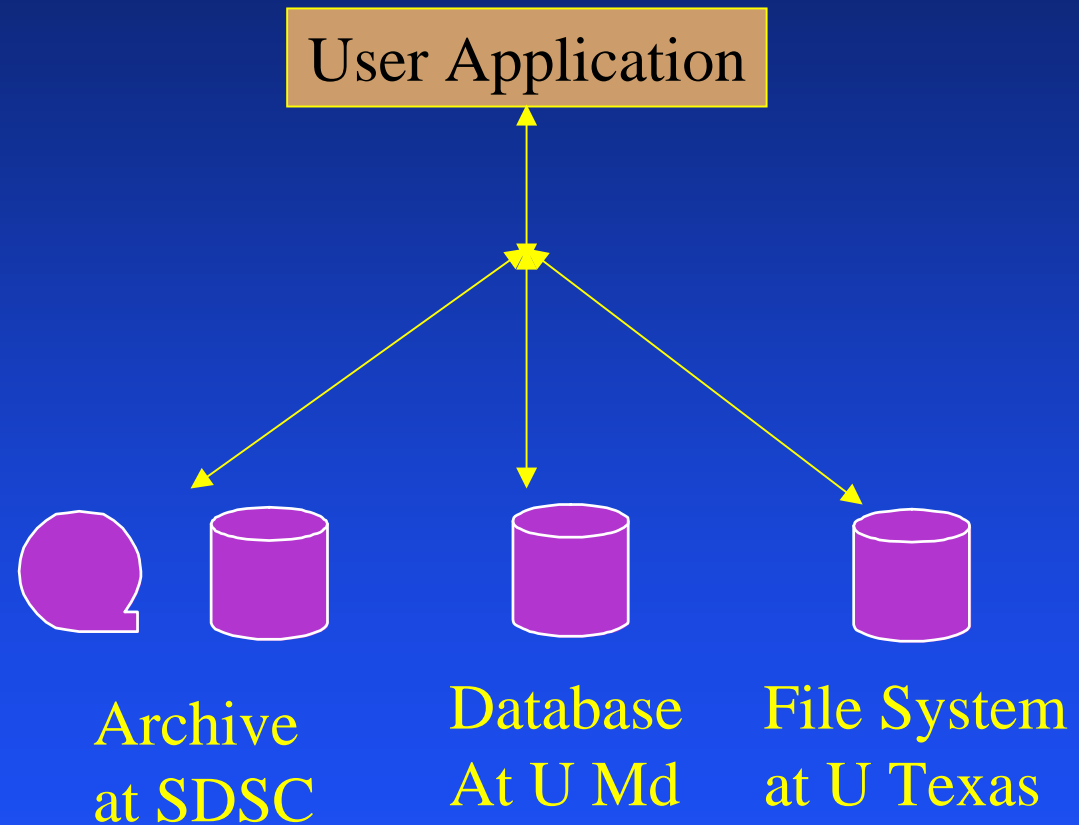
Third party transfer

Filtering

Queries



# Data Virtualization



# Data Virtualization

Logical name space

Location independent identifier

Persistent identifier

Collection owned data

Access controls

Audit trails

Checksums

Descriptive metadata

Inter-realm authentication

Single sign-on system

User Application

Common naming convention and set of attributes for describing digital entities



Archive  
at SDSC

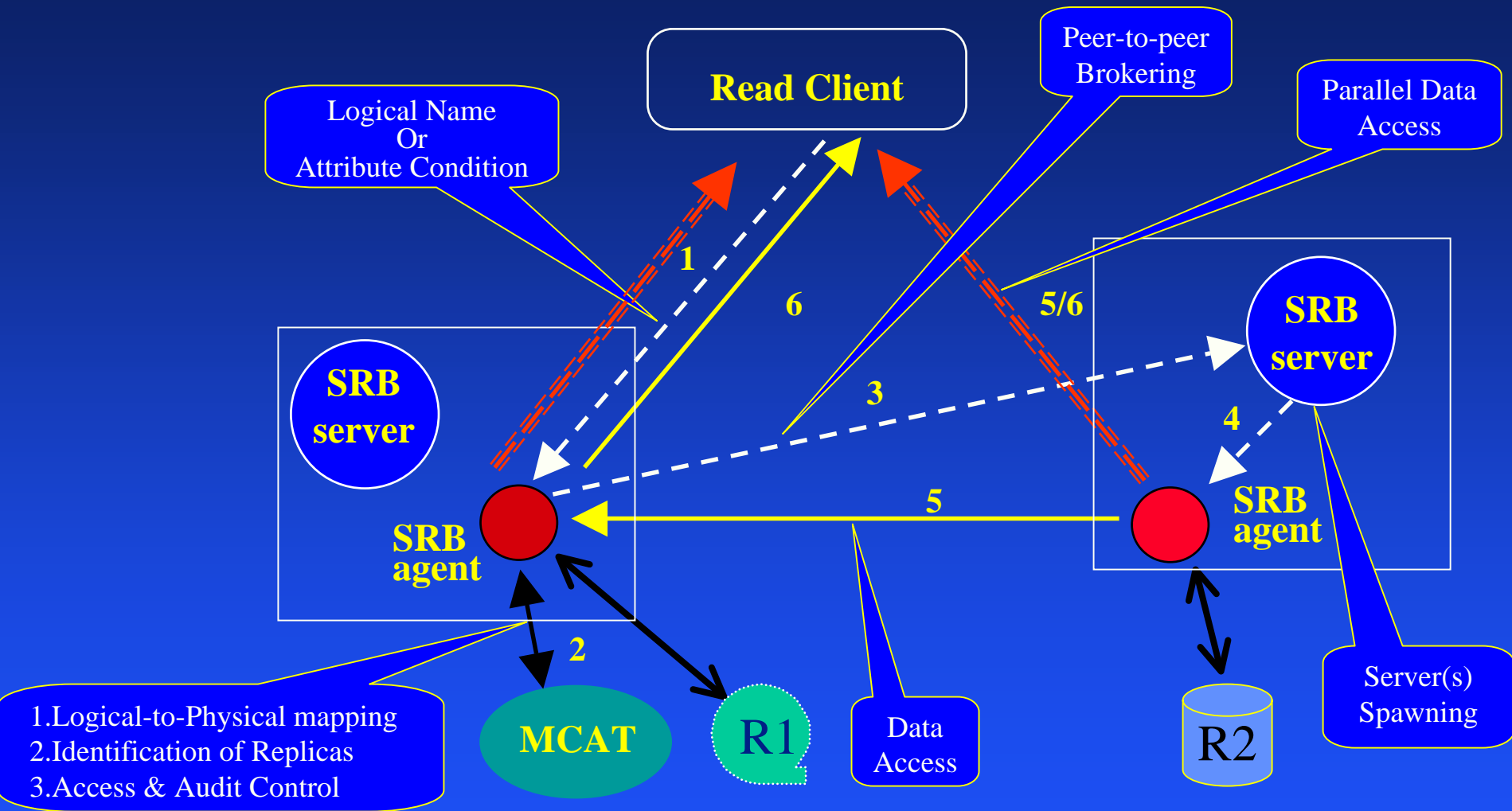
Database  
At U Md

File System  
at U Texas

# Three Tier Architecture

- **Clients**
  - Your preferred access mechanism
- **Metadata catalog**
  - Separation of metadata management from data storage
- **Servers**
  - Manage interactions with storage systems
  - Federated to support direct interactions between servers

# Federated SRB server model





# SDSC Storage Resource Broker & Meta-data Catalog

**Application**

**C, C++,  
Libraries**

**Linux  
I/O**

**Unix  
Shell**

**Java, NT  
Browsers**

**DLL /  
Python**

**GridFTP**

**OAI  
WSDL**

**Access  
APIs**

**Consistency Management / Authorization-Authentication**

**Logical Name  
Space**

**Latency  
Management**

**Data  
Transport**

**Metadata  
Transport**

**SRB  
Server**

**Catalog Abstraction**

**Storage Abstraction**

**Databases  
DB2, Oracle, Sybase,  
SQLServer**

**Archives  
HPSS, ADMS,  
UniTree, DMF**

**HRM**

**File Systems  
Unix, NT,  
Mac OSX**

**Databases  
DB2, Oracle,  
Postgres**

**Drivers**



# SRB Name Spaces

- **Digital Entities (files, blobs, Structured data, ...)**
  - Logical name space for files for global identifiers
- **Resources**
  - Logical names for managing collections of resources
- **User names (user-name / domain / SRB-zone)**
  - Distinguished names for users to manage access controls
- **MCAT metadata**
  - Standard metadata attributes, Dublin Core, administrative metadata

# Logical Name Space

- **Global, location-independent identifiers for digital entities**
  - Organized as collection hierarchy
  - Attributes mapped to logical name space
    - Attributed managed in a database
- **Types of administrative metadata**
  - Physical location of file
  - Owner, size, creation time, update time
  - Access controls

# Data Identifier Transparency

## Four Types of Data Identifiers:

- **Unique name**
  - OID or handle
- **Descriptive name**
  - Descriptive attributes – meta data
  - Semantic access to data
- **Collective name**
  - Logical name space of a collection of data sets
  - Location independent
- **Physical name**
  - Physical location of resource and physical path of data

# Mappings on Resource Name Space

- **Define logical resource name**
  - List of physical resources
- **Replication**
  - Write to logical resource completes when all physical resources have a copy
- **Load balancing**
  - Write to a logical resource completes when copy exist on next physical resource in the list
- **Fault tolerance**
  - Write to a logical resource completes when copies exist on “k” of “n” physical resources

# Data Replica Transparency

- **Replication**

- Improve access time
- Improve reliability
- Provide disaster backup and preservation
- Physically or Semantically equivalent replicas

- **Replica consistency**

- Synchronization across replicas on writes
- Updates might use “m of n” or any other policy
- Distributed locking across multiple sites

- **Versions of files**

- Time-annotated snapshots of data

# Latency Management -Bulk Operations

- **Bulk register**
  - Create a logical name for a file
- **Bulk load**
  - Create a copy of the file on a data grid storage repository
- **Bulk unload**
  - Provide containers to hold small files and pointers to each file location
- **Bulk delete**
  - Mark as deleted in metadata catalog
  - After specified interval, delete file
- **Bulk metadata load**
- **Requests for bulk operations for access control setting, ...**

# SRB Latency Management

Remote Proxies,  
Staging

Data Aggregation  
Containers

Prefetch



Replication  
Server-initiated I/O

Streaming  
Parallel I/O

Caching  
Client-initiated I/O



# Remote Proxies

- **Extract image cutout from Digital Palomar Sky Survey**
  - Image size 1 Gbyte
  - Shipped image to server for extracting cutout took 2-4 minutes (5-10 Mbytes/sec)
- **Remote proxy performed cutout directly on storage repository**
  - Extracted cutout by partial file reads
  - Image cutouts returned in 1-2 seconds
- **Remote proxies are a mechanism to aggregate I/O commands**

# Virtual Data Abstraction

- **Virtual Data or “*On Demand Data*”**
  - Created on demand is not already available
  - *Recipe* to create derived data
  - Grid based computation to create derived data product
- **Object based storage (extended data operations)**
  - Data subsetting at the remote storage repository
  - Data formatting at the remote storage repository
  - Metadata extraction at the remote storage repository
  - Bulk data manipulation at the remote storage repository

# Grid Bricks

- **Integrate data management system, data processing system, and data storage system into a modular unit**
  - Commodity based disk systems (1 TB)
  - Memory (1 GB)
  - CPU (1.7 Ghz)
  - Network connection (Gig-E)
  - Linux operating system
- **Data Grid technology to manage name spaces**
  - User names (authentication, authorization)
  - File names
  - Collection hierarchy

# Data Grid Brick

- **Hardware components**
  - Intel Celeron 1.7 GHz CPU
  - SuperMicro P4SGA PCI Local bus ATX mainboard
  - 1 GB memory (266 MHz DDR DRAM)
  - 3Ware Escalade 7500-12 port PCI bus IDE RAID
  - 10 Western Digital Caviar 200-GB IDE disk drives
  - 3Com Etherlink 3C996B-T PCI bus 1000Base-T
  - Redstone RMC-4F2-7 4U ten bay ATX chassis
  - Linux operating system
- **Cost is \$2,200 per Tbyte plus tax**
- **Gig-E network switch costs \$500 per brick**
- **Effective cost is about \$2,700 per TByte**

# Grid Bricks at SDSC

- **Used to implement “picking” environments for 10-TB collections**
  - Web-based access
  - Web services (WSDL/SOAP) for data subsetting
- **Implemented 15-TBs of storage**
  - Astronomy sky surveys, NARA prototype persistent archive, NSDL web crawls
- **Must still apply Linux security patches to each Grid Brick**
- **Grid bricks managed through SRB**
  - Logical name space, User Ids, access controls
  - Load leveling of files across bricks

# Data Grid Federation

- Data grids provide the ability to name, organize, and manage data on distributed storage resources
- Federation provides a way to name, organize, and manage data on multiple data grids.

# SRB Zones

- Each SRB zone uses a metadata catalog (MCAT) to manage the context associated with digital content
- Context includes:
  - Administrative, descriptive, authenticity attributes
  - Users
  - Resources
  - Applications

# SRB Peer-to-Peer Federation

- **Mechanisms to impose consistency and access constraints on:**
  - Resources
    - Controls on which zones may use a resource
  - User names (user-name / domain / SRB-zone)
    - Users may be registered into another domain, but retain their home zone, similar to Shibboleth
  - Data files
    - Controls on who specifies replication of data
  - MCAT metadata
    - Controls on who manages updates to metadata



# Peer-to-Peer Federation

1. Occasional Interchange - for specified users
2. Replicated Catalogs - entire state information replication
3. Resource Interaction - data replication
4. Replicated Data Zones - no user interactions between zones
5. Master-Slave Zones - slaves replicate data from master zone
6. Snow-Flake Zones - hierarchy of data replication zones
7. User / Data Replica Zones - user access from remote to home zone
8. Nomadic Zones “SRB in a Box” - synchronize local zone to parent
9. Free-floating “myZone” - synchronize without a parent zone
10. Archival “BackUp Zone” - synchronize to an archive

SRB Version 3.0.1 released December 19, 2003

# Principle peer-to-peer federation approaches (1536 possible combinations)

Zone SRB	Zone Organization	Zone interaction control	Consistency Management	User Connection Point to access files	Data Access Control Setting	Metadata synchroni- zation	Resource sharing	User-ID sharing between zones
	Zones	Zones	Collections	Files	Files	Metadata	Resources	User names
<b>Free Floating Zones</b>	Peer-to-Peer	Local Admin	User-specified data publication	From home zone	User set access controls	User controlled synchronization	None	None
<b>Occasional Interchange</b>	Peer-to-Peer	Local Admin	User specified	From home zone	User set access controls	User controlled synchronization	None	Partial
<b>Replicated Data Zones</b>	Peer-to-Peer	Local Admin	User-specified replication	From home zone	User set local access controls	User controlled synchronization	Partial	Partial, user establishes own accounts
<b>Resource Interaction</b>	Peer-to-Peer	Local Admin	User-specified replication	From home zone	User set access controls	None	Partial shared resource for replication	Partial
<b>User and Data Replica Zones</b>	Peer-to-Peer	Local Admin	User-specified replication	From home zone	System set access controls	System controlled complete synchronization	Partial	Complete
<b>Replicated Catalog</b>	Peer-to-Peer	Local Admin	System managed name conflict resolution	From any zone	System replicated access controls	System controlled complete synchronization	All zones share resources	Complete
<b>Snow Flake Zones</b>	Hierarchical	Local Admin	System managed replication in hierarchy of zones	From home zone	System set access controls	System controlled partial synchronization	None	One
<b>Master-Slave Zones</b>	Hierarchical	Super Admin	System-managed replication to slave	From home zone	System set access controls	System controlled partial synchronization	None	One
<b>Archival zones</b>	Hierarchical	Super Admin	System-managed versioning to parent zone	From home zone	System set access controls	System controlled complete synchronization	None	Complete
<b>Nomadic Zones</b>	Hierarchical	Local Admin	User-managed replication to parent zone	From home zone	User set access controls	User controlled synchronization	Partial	One

# Comparison of peer-to-peer federation approaches

Free Floating

Partial User-ID Sharing

Occasional Interchange

Partial Resource Sharing

Replicated Data

System Set Access Controls  
System Controlled Complete Synch  
Complete User-ID Sharing

User and Data Replica

System Managed Replication  
Connection From Any Zone  
Complete Resource Sharing

Replicated Catalog

No Metadata Synch

Resource Interaction

Hierarchical Zone Organization  
One Shared User-ID

Nomadic

System Managed Replication  
System Set Access Controls  
System Controlled Partial Metadata Synch  
No Resource Sharing

Snow Flake

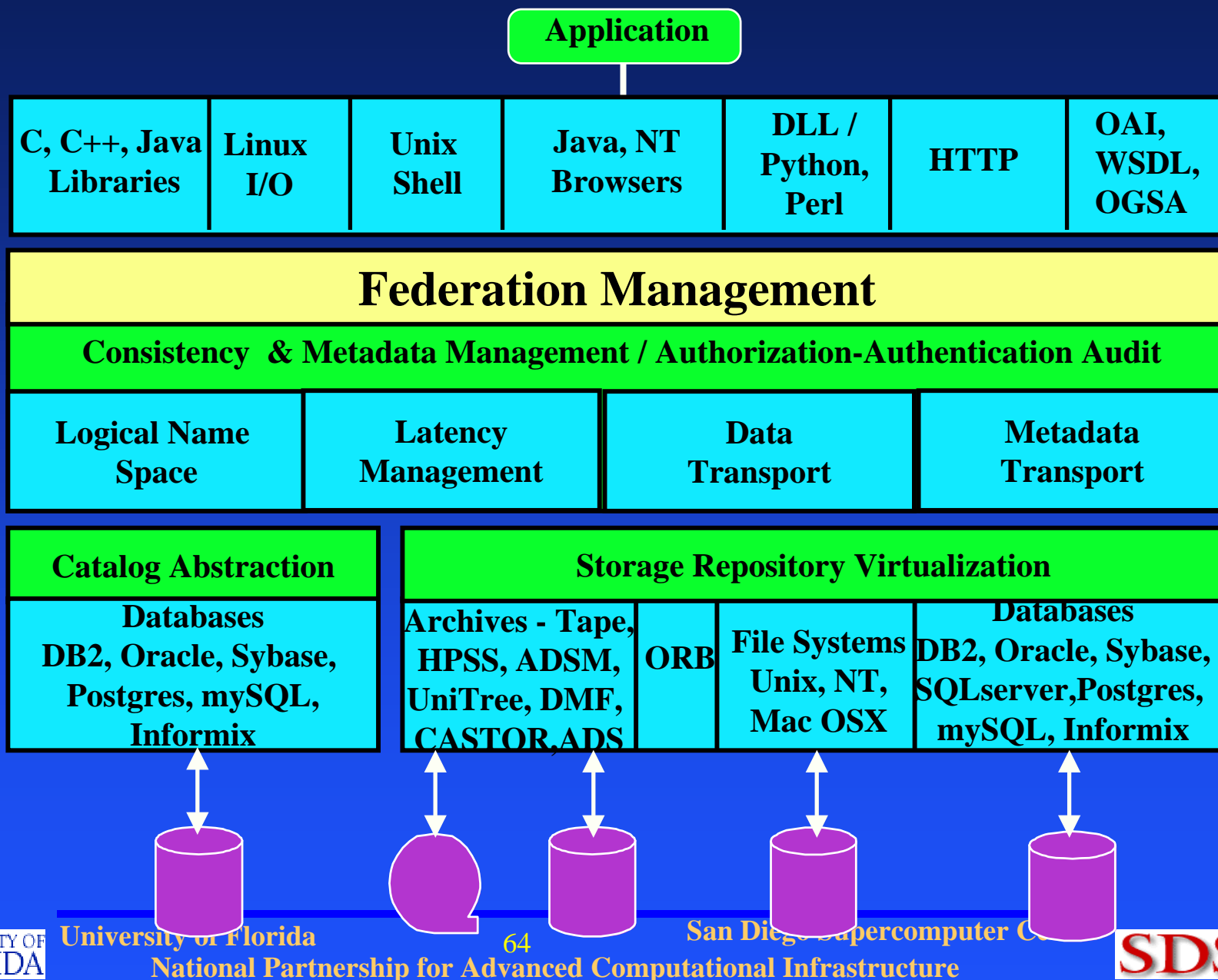
Super Administrator Zone Control

Master Slave

System Controlled Complete Metadata Synch  
Complete User-ID Sharing

Archival

# Data Grid Federation - zoneSRB



# Data Organization

- **Physical Organization of the data**
  - Distributed Data
  - Heterogeneous resources
  - Multiple formats (structured and unstructured)
- **Logical Organization**
  - Impose logical structure for data sets
  - *Collections* of semantically related data sets
  - Users create their own views (collections) of the data grid
- **Digital Ontology**
  - Characterization of structures in data sets and collections
  - Mapping of semantic labels to the structures

# Tutorial Outline

- **Introduction**
  - Data Grids
  - Data Grid Infrastructures
- **Information Management using Data Grids**
  - Data Grid Transparencies and concepts
  - Peer-to-peer Federation of Data Grids
- ➡ **Gridflows and Data Grids**
  - Need for Gridflows
  - Data Grid Language and SDSC Matrix Project
- **Data Grids and You**
  - Open Research Issues and Global Grid Forum Community
- **Lets build a Data Grid**
  - Using SDSC SRB Data Grid Management System and its Interfaces

# Gridflows

- **Grid Workflow (Gridflow) is the automation of a execution pipeline in which data or tasks are processed through multiple autonomous grid resources according to a set of procedural rules**
- **Gridflows are executed on resources that are dynamically obtained through confluence of one or more autonomous administrative domains (peers)**

[illegible]



# DG-Builder to create Gridflows

The screenshot displays the DG-Builder 0.1 application window. The main workspace shows a flow diagram with a green box labeled 'DataGrid Request' connected to an orange oval labeled 'Flow myFirstFlow'. A dashed arrow points from the flow to a blue box labeled 'Step: ChageStep1'. A right-hand pane titled 'Properties' shows a table with 'Property' and 'Value' columns, containing 'Flow' and 'Flow ID' with the value 'Flow 123 xyz'. A bottom pane shows the XML code for the flow.

```

<keyvalue>4545</keyvalue>
</key>
</ticketID>
</gridTicket>
</GridUser>
- <VOInfo>
  <voID>SCEC</voID>
</VOInfo>
- <Flow flowID="myFirstFlow">
  - <flowLogic>
    - <forEachLoop>
      - <iterationSet>
        - <collection>
          - <StdCollection>
            <collectionName>/home/matrix.sdsc/testCollection</collectionName>
          </StdCollection>
        </collection>
      </iterationSet>
      <iterationVariableName>dataSet</iterationVariableName>
    </forEachLoop>
  </flowLogic>
  - <Steps stepID="ChageStep1">
    - <Operation>
      - <ChangePermissionOp>
        - <StdParams>
          - <gridData>
            - <DataSetNow>
              <dataIdentifier>$dataSet</dataIdentifier>
            </DataSetNow>
          </gridData>
          - <targetUser>
            - <gridTicket>
              - <ticketID>
                - <key>

```

# Need for Gridflows

- **Data-intensive and/or compute-intensive processes**
  - Long run processes or pipelines on the Grid
  - (e.g) If job A completes execute jobs x, y, z; else execute job B.
- **Self-organization/management of data**
  - Semi-automation of data, storage distribution, curation processes
  - (e.g) After each data insert into a collection, update the meta-data information about the collection or replicate the collection
- **Knowledge Generation**
  - Offline data analysis and knowledge generation pipelines
  - (e.g) What inferences can be assumed from the new seismology graphs added to this collection? Which domain scientist will be interested to study these new possible pre-results?

# Gridflow Description Requirements

- **Import and export**
  - Import or export Gridflows (embedded gridflows)
  - Support and extend existing standards like XQuery, BPEL, SOAP etc.,
- **Rules**
  - Dynamic rules to control the execution of gridflow
- **Query**
  - Runtime Query on status of gridflow
- **Granular Metadata**
  - Metadata associated with the steps in a gridflow execution that can be queried
- **Gridflow Patterns**
  - Scientific Computing - more looping structures
  - Interest in execution of each iteration and the changes in interested attributes
  - <http://tmitwww.tm.tue.nl/research/patterns/standards.htm>

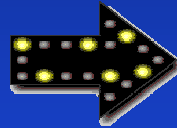
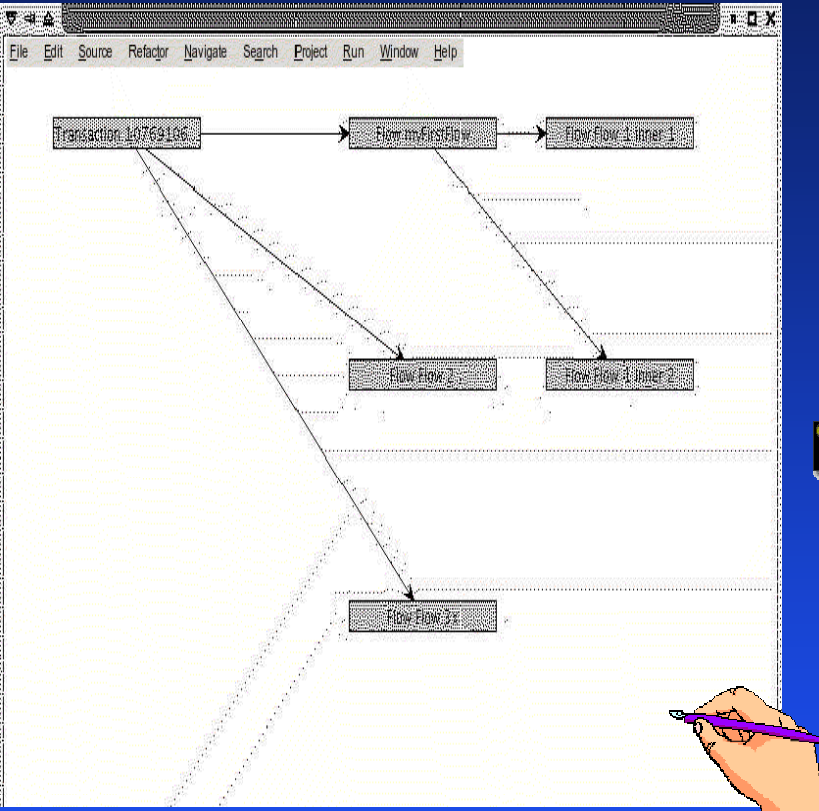
# Data Grid Language

- *Assembly Language for Grid Computing*
- **Describes Gridflow**
  - Both structure-based and state-based gridflow patterns
  - Described ECA based rules
  - Inbuilt support to define data grid datatypes like collections, ...
- **Query Gridflow**
  - Query on the execution of any gridflow (any granular detail)
  - XQuery is used to query on the status of gridflow and its attributes
- **Manage Gridflow**
  - Start or stop the gridflow in execution

# Structure and state based Gridflow patterns

- **Simple Sequential**
  - Execute steps in a gridflow in a sequence one after another
- **Simple Parallel**
  - Start all the steps in a gridflow at the same time
- **For Loop Iteration**
  - Execute steps changing some iterator value until a given state is achieved
- **While Block (Milestone)**
  - Execute steps while some mile stone can be achieved
- **IF-Else Block**
  - Branch based on the evaluation of a state condition
- **Switch-choice(s)**
  - Split to execute any of the possible cases based on the context
- **More.. (For-each, BPEL etc)**

# Gridflow Process I



```

<flowLogic>
- <Flow flowID="forFlow">
- <flowLogic>
- <forLoop>
- <init name="optional">
  <actionString>SET THIS.i = 0</actionString>
</init>
- <forCondition>
  <xQuery>number
    (#THIS.DGL#/TransactionStatusResponse/FlowStatusResponse
    [@ID="forFlow"]/Variables/part[@name = "i"]//text()) < 2</xQuery>
  </forCondition>
- <iterator name="optionalThingie">
  <documentation>arithmetic operation of meta-data</documentation>
  <actionString>SET THIS.i = THIS.i + 1</actionString>
</iterator>
</forLoop>
- <userDefinedRules name="beforeEntry">
- <condition>
  <xQuery>#THIS.DGL#/TransactionStatusResponse/Variables/part[@name =
  "color"]//text() = "green"</xQuery>
</condition>
- <actions name="true">
  <actionString>SET THIS.newVar = "Matrix"</actionString>
</actions>
</userDefinedRules>
</flowLogic>
- <Steps stepID="forStep1">
- <UserDefinedRules name="beforeEntry">
- <condition>
  <documentation>Text</documentation>
  <xQuery>"green"</xQuery>
</condition>
- <actions name="green">
  <documentation>Text</documentation>

```

End User using DGBuilder

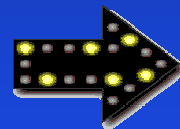
Gridflow Description  
Data Grid Language

# Gridflow Process II

```

<Flow flowID="forFlow">
  <flowLogic>
    <forLoop>
      <init name="optional">
        <actionString>SET THIS.i = 0</actionString>
      </init>
      <forCondition>
        <xQuery>number
          (#THIS.DGL#/TransactionStatusResponse/FlowStatusResponse
            [@ID="forFlow"]/Variables/part[@name = "i"]//text()) < 2</xQuery>
      </forCondition>
      <iterator name="optionalThingie">
        <documentation>arithmetic operation of meta-data</documentation>
        <actionString>SET THIS.i = THIS.i + 1</actionString>
      </iterator>
    </forLoop>
    <userDefinedRules name="beforeEntry">
      <condition>
        <xQuery>#THIS.DGL#/TransactionStatusResponse/Variables/part[@name =
          "color"]//text() = "green"</xQuery>
      </condition>
      <actions name="true">
        <actionString>SET THIS.newVar = "Matrix"</actionString>
      </actions>
    </userDefinedRules>
  </flowLogic>
  <Steps stepID="forStep1">
    <UserDefinedRules name="beforeEntry">
      <condition>
        <documentation>Text</documentation>
        <xQuery>"green"</xQuery>
      </condition>
      <actions name="green">
        <documentation>Text</documentation>

```



```

<Symbol>List</Symbol>
<Function>
  <Symbol>List</Symbol>
  <Symbol>Automatic</Symbol>
  <Number>4</Number>
</Function>
<Function>
  <Symbol>List</Symbol>
  <Symbol>Automatic</Symbol>
  <Number>6</Number>
</Function>
</Options>
</Notebook>

```

```

<Symbol>List</Symbol>
<Function>
  <Symbol>List</Symbol>
  <Symbol>Automatic</Symbol>
  <Number>4</Number>
</Function>
<Function>
  <Symbol>List</Symbol>
  <Symbol>Automatic</Symbol>
  <Number>6</Number>
</Function>
</Options>
</Notebook>

```

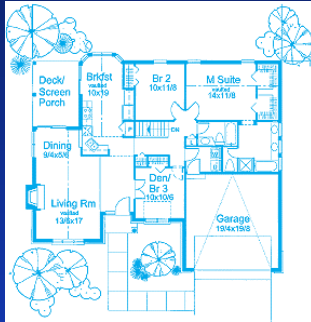
Abstract Gridflow using  
Data Grid Language

Planner

Concrete Gridflow



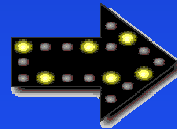
# Gridflow Process III



```
<Symbol>List</Symbol>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>4</Number>
</Function>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>6</Number>
</Function>
</Option>
</Options>
</Notebook>
```

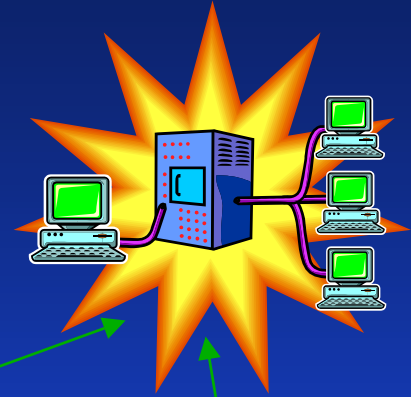
```
<Symbol>List</Symbol>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>4</Number>
</Function>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>6</Number>
</Function>
</Option>
</Options>
</Notebook>
```

Gridflow Processor

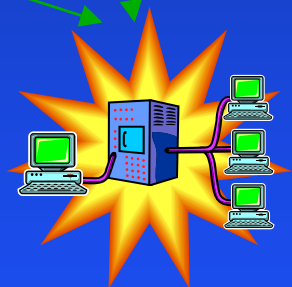


Concrete Gridflow

```
<Symbol>List</Symbol>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>4</Number>
</Function>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>6</Number>
</Function>
</Option>
</Options>
</Notebook>
```



```
<Symbol>List</Symbol>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>4</Number>
</Function>
<Function>
<Symbol>List</Symbol>
<Symbol>Automatic</Symbol>
<Number>6</Number>
</Function>
</Option>
</Options>
</Notebook>
```



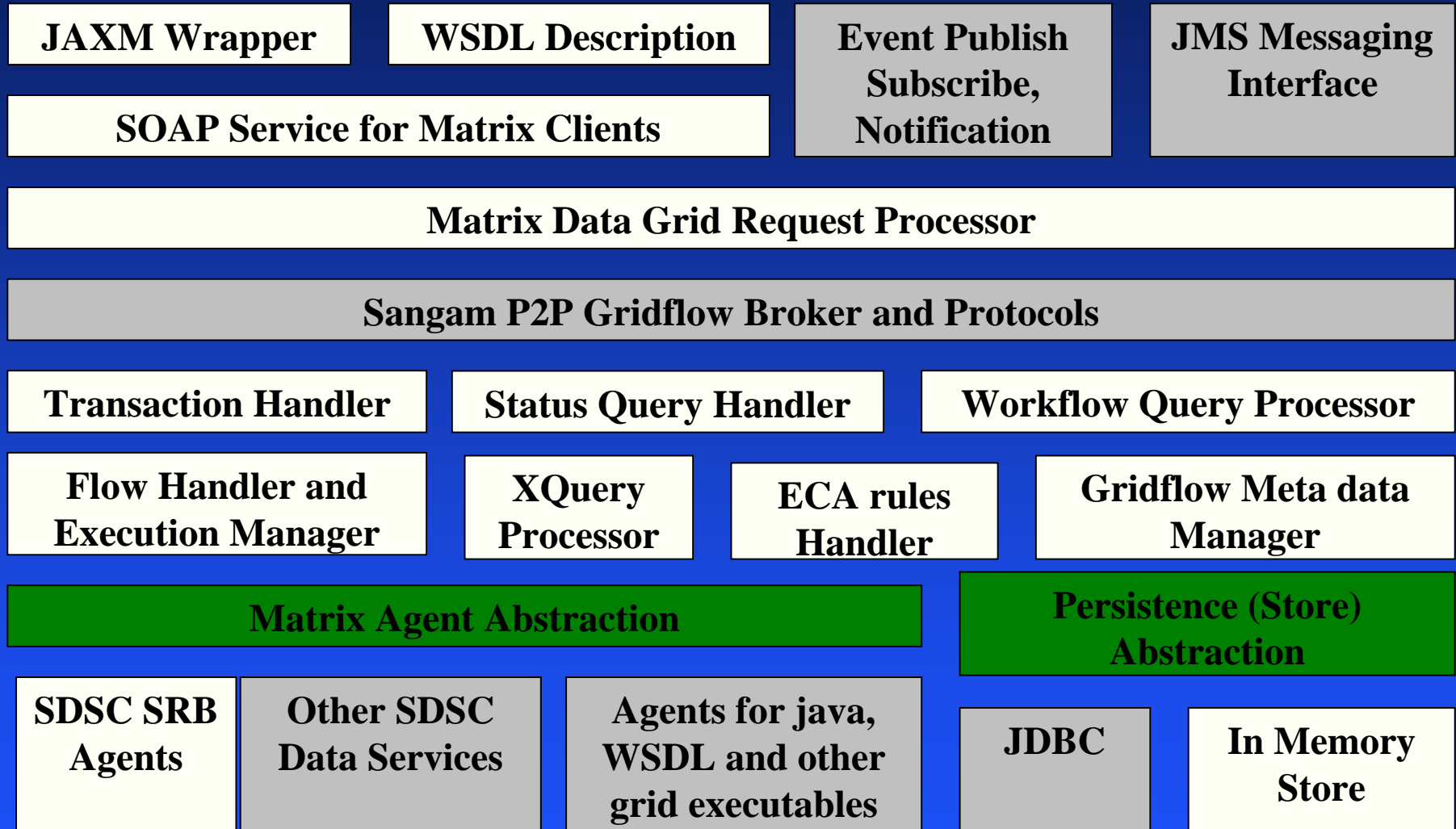
Gridflow P2P Network



# SDSC Matrix Project

- **R&D effort that is ready for production now**
  - Gridflow Protocols
  - Gridflow Language Descriptions
  - Version 3.0 released
- **Community based**
  - Apache Software License
  - Both Industry and Academia can benefit by participation
  - Involves University of Florida, UCSD, ... (Are you In?)
- **Multiple Projects could be benefited**
  - Very large academic data grid projects
  - Industries which want to be the early adopters

# Matrix Gridflow Server Architecture



# Matrix Gridflow System Features

- **Support of Data Grid Language**
  - Both state-based and structure-based gridflow branching
  - Working on BPEL integration
  - Scoped meta-data variables useful for tracking the state
  - Status Queries at run-time
- **Gridflow provenance tracking**
  - Inbuilt database support that can track all activities in your Grid
- **End-user GUI**
  - Users would be able to click and drag/draw gridflow graphs
  - DG-Builder to be release in the first week of April

# **SDSC Matrix Project:**

## **Open source effort by SDSC and SRB folks**

- The growth of the SDSC Matrix Project is made possible by developers and grid-prophets like you (Thank you)
- [talk2Matrix@sdsc.edu](mailto:talk2Matrix@sdsc.edu)

# Tutorial Outline

- **Introduction**
  - Data Grids
  - Data Grid Infrastructures
- **Information Management using Data Grids**
  - Data Grid Transparencies and concepts
  - Peer-to-peer Federation of Data Grids
- **Gridflows and Data Grids**
  - Need for Gridflows
  - Data Grid Language and SDSC Matrix Project
- ➡ **Data Grids and You**
  - Open Research Issues and Global Grid Forum Community
- **Lets build a Data Grid**
  - Using SDSC SRB Data Grid Management System and its Interfaces

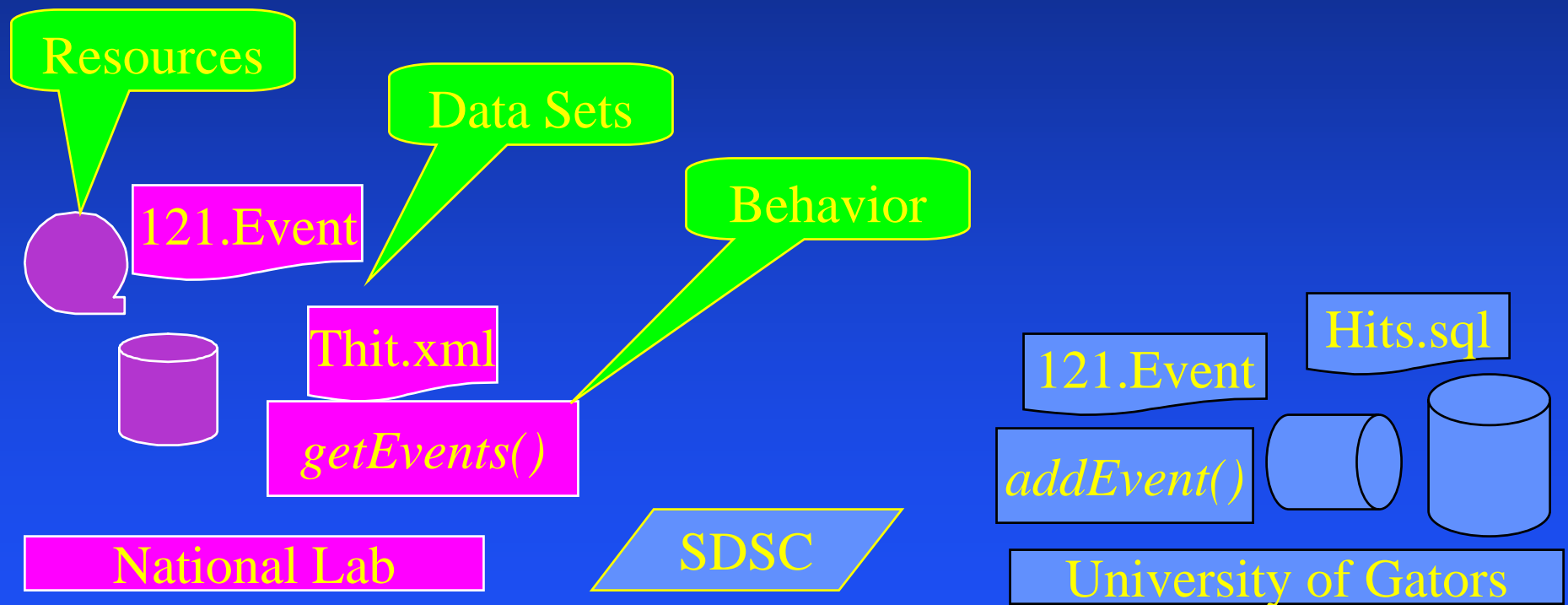
# DGMS Philosophy

- **Collective view of**
  - Inter-organizational data
  - Operations on datagrid space
- **Local autonomy and global state consistency**
- **Collaborative datagrid communities**
  - Multiple administrative domains or “Grid Zones”
- **Self-describing and self-manipulating data**
  - Horizontal and vertical behavior
  - Loose coupling between data and behavior (dynamically)
  - Relationships between a digital entity and its *Physical locations, Logical names, Meta-data, Access control, Behavior, “Grid Zones”*.

# DGMS Research Issues

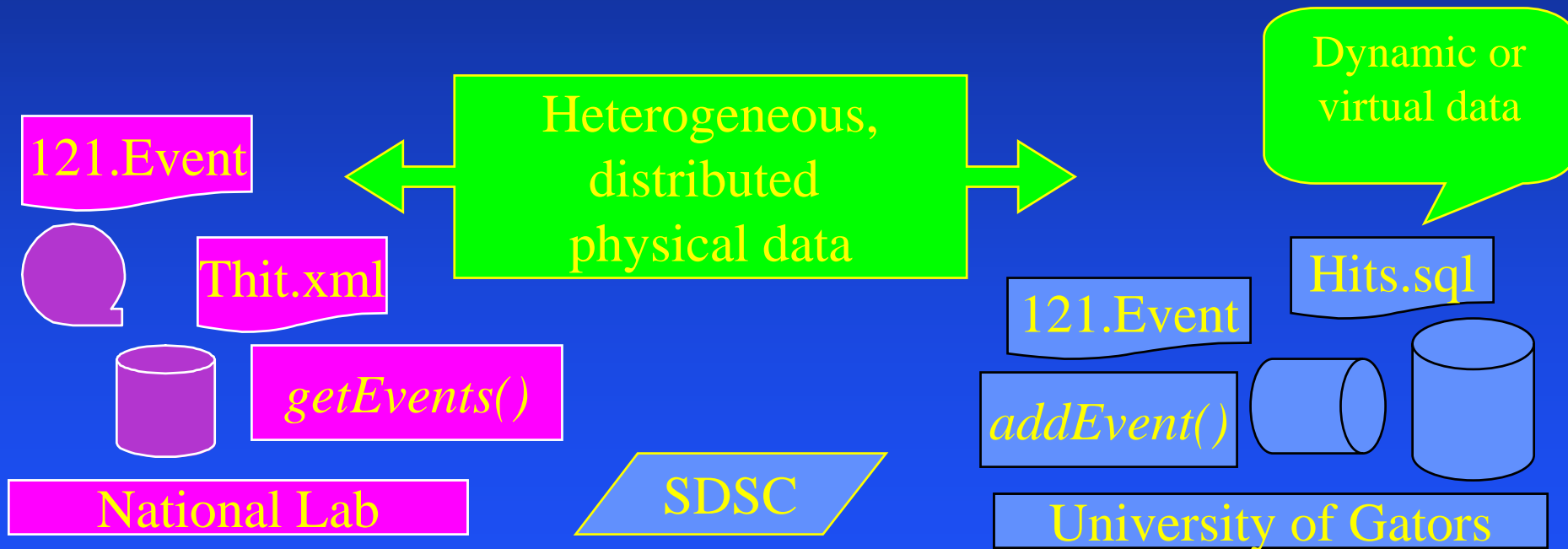
- **Self-organization of datagrid communities**
  - Using knowledge relationships across the datagrids
  - Inter-datagrid operations based on semantics of data in the communities (different ontologies)
- **High speed data transfer**
  - Terabyte to transfer - TCP/IP not final answer
  - Protocols, routers needed
- **Latency Management**
  - Data source speed  $\gg$  data sink speed
- **Datagrid Constraints**
- **Data placement and scheduling**
  - How many replicas, where to place them...

# Active Datagrid Collections

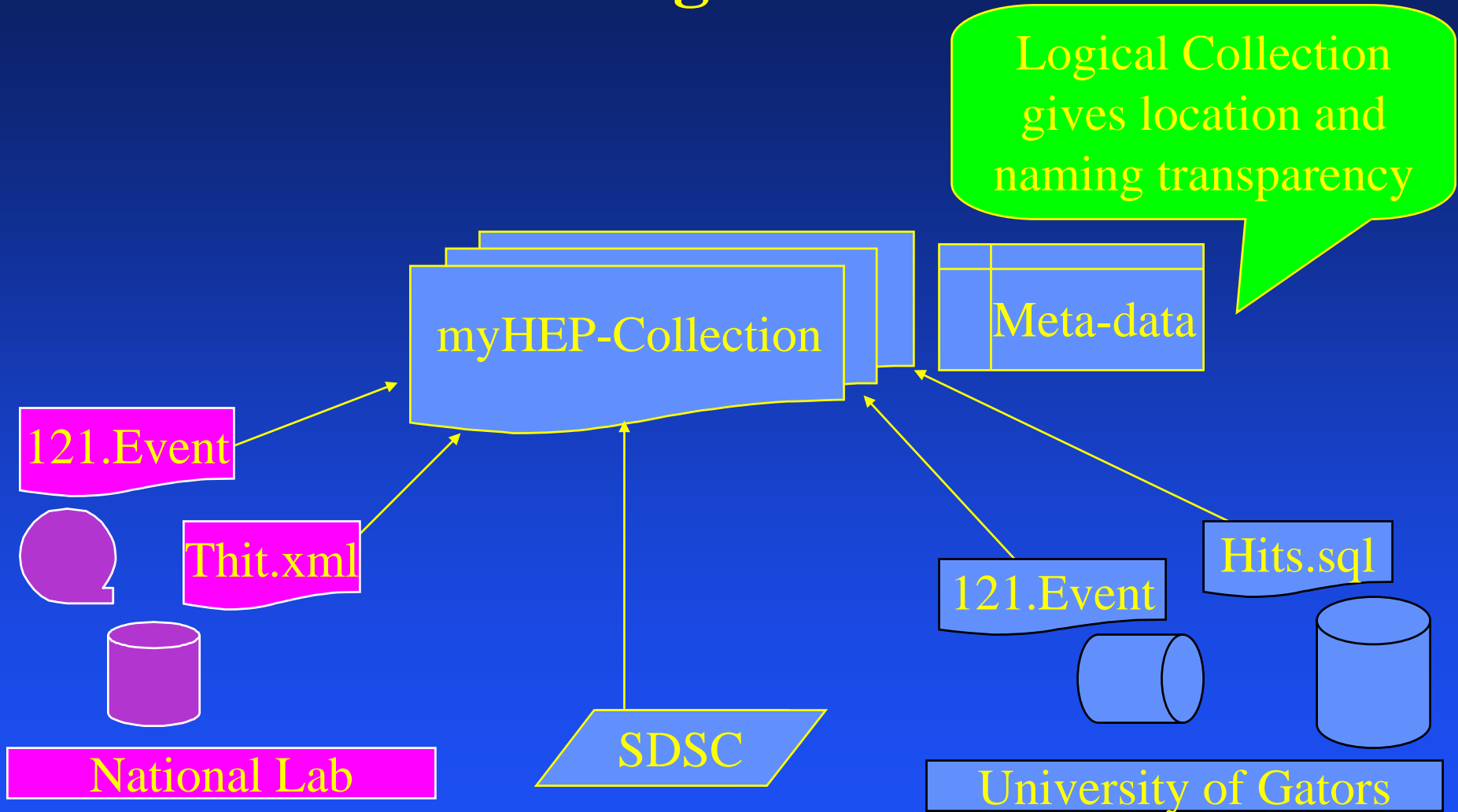




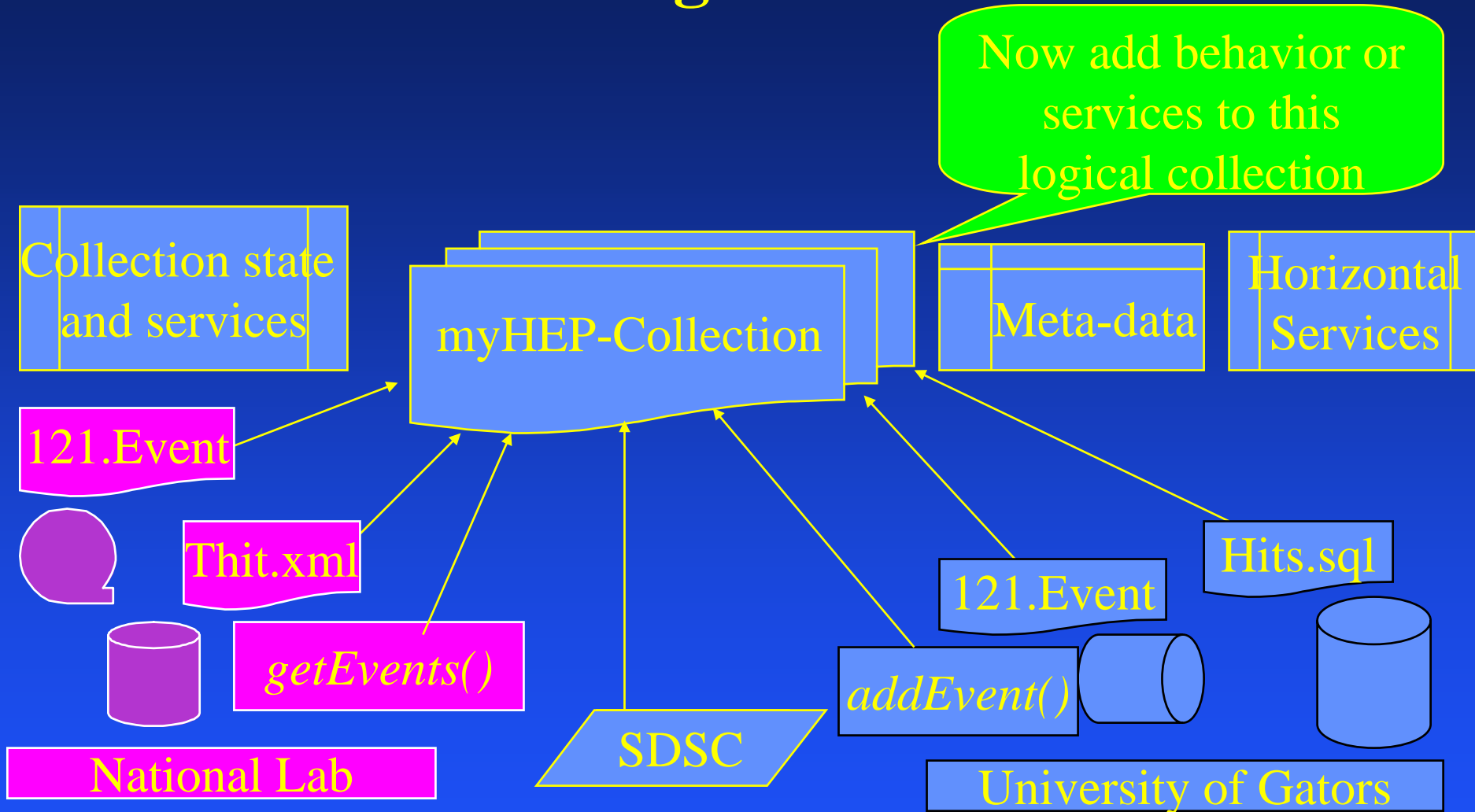
# Active Datagrid Collections



# Active Datagrid Collections



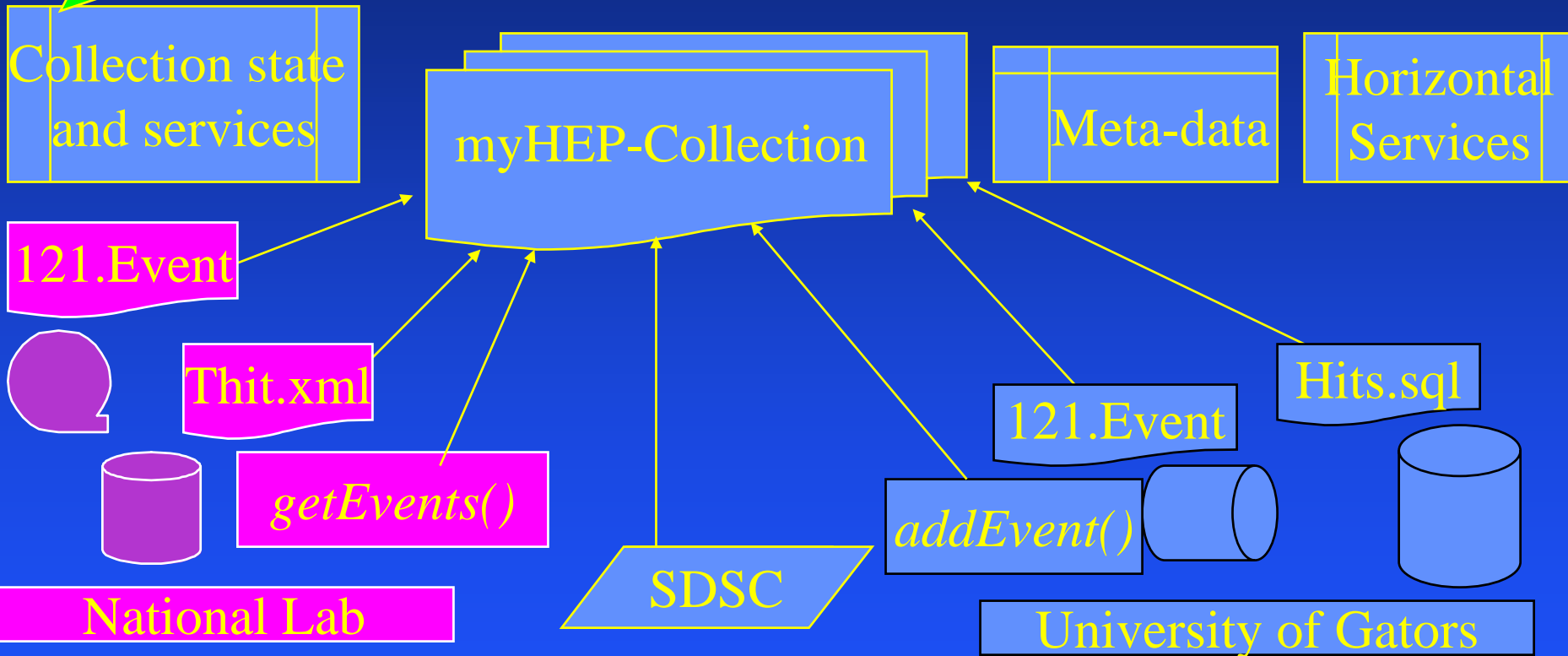
# Active Datagrid Collections



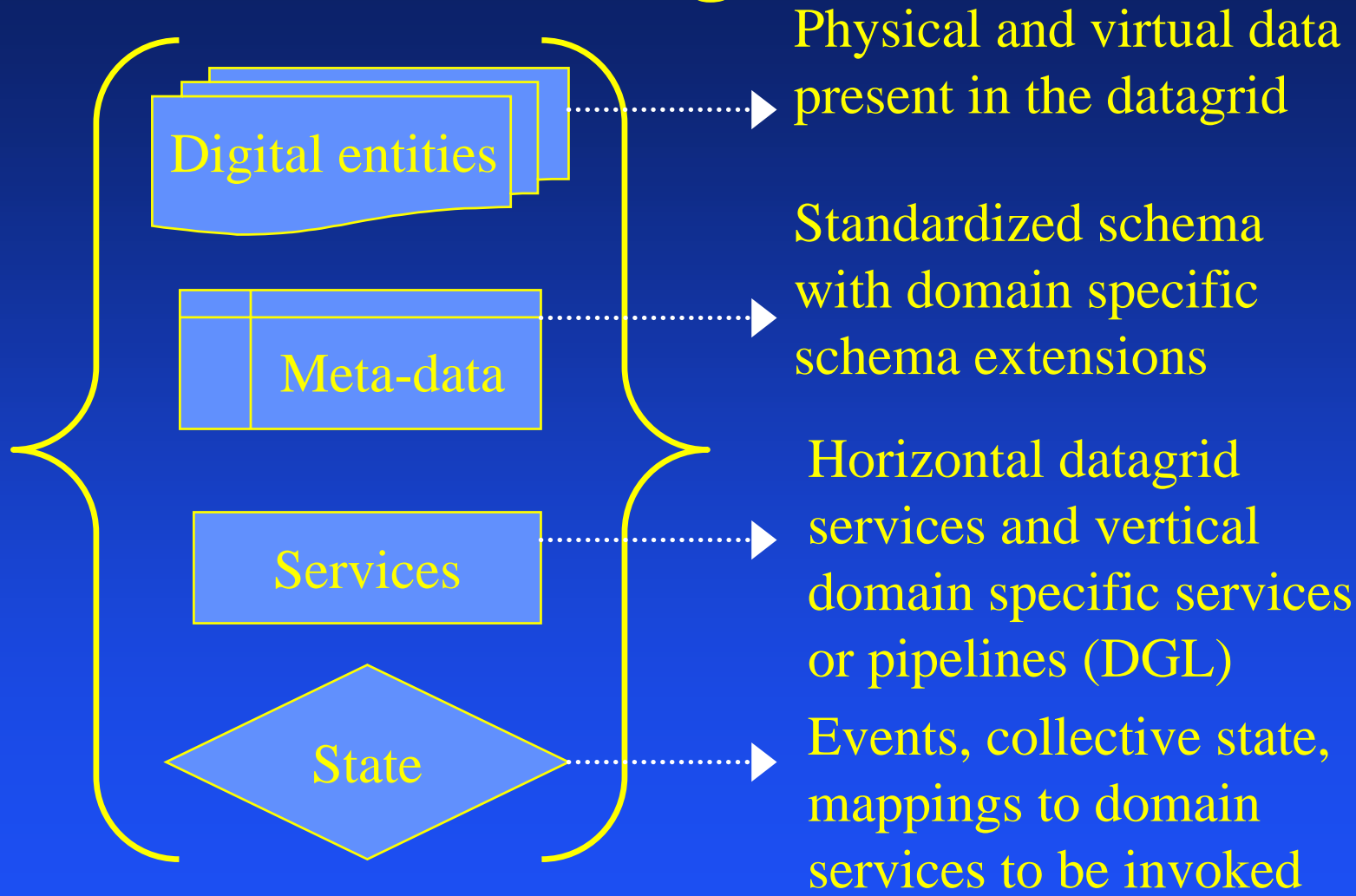
# Active Datagrid Collections

ADC specific  
Operations + Model View  
Controllers

ADC Logical view  
of data & operations



# Active Datagrid Collections



# Global Grid Forum (GGF)

- **Global Forum for Information Exchange and Collaboration**
  - Promote and support the development and deployment of Grid Technologies
  - Creation and documentation of “best practices”, technical specifications (standards), user experiences, ...
  - Modeled after Internet Standards Process (IETF, RFC 2026)
  - <http://www.ggf.org>

# Tutorial Outline

- **Introduction**
    - Data Grids
    - Data Grid Infrastructures
  - **Information Management using Data Grids**
    - Data Grid Transparencies and concepts
    - Peer-to-peer Federation of Data Grids
  - **Gridflows and Data Grids**
    - Need for Gridflows
    - Data Grid Language and SDSC Matrix Project
  - **Data Grids and You**
    - Open Research Issues and Global Grid Forum Community
- ➡ **Lets build a Data Grid**
- Using SDSC SRB Data Grid Management System and its Interfaces

# SRB Information Resources

- **SRB Homepage:**
  - <http://www.npaci.edu/DICE/SRB/>
- **inQ Homepage**
  - <http://www.npaci.edu/dice/srb/inQ/inQ.html>
- **mySRB URL**
  - <https://srb.npaci.edu/mySRB2v7.shtml>
- **Grid Port Toolkit**
  - <https://gridport.npaci.edu/>
- **SRB Chat**
  - [srb-chat@sdsc.edu](mailto:srb-chat@sdsc.edu)
- **SRB bug list**
  - <http://www.npaci.edu/dice/srb/bugs.html>



# SRB Availability

- SRB source distributed to academic and research institutions
- Commercial use access through UCSD Technology Transfer Office
  - William Decker [WJDecker@ucsd.edu](mailto:WJDecker@ucsd.edu)
- Commercial version from
  - <http://www.nirvanastorage.com>

# SRB Production

- **Goal is to eliminate all known bugs**
- **Major releases every year (1.0, 2.0, 3.0)**
  - Provide major new capabilities
- **Minor releases (2.1, 2.2)**
  - Provide upgrades, ports, bug fixes
- **Bug fix releases (2.1.1)**
  - Specific releases to fix urgent problems at a given site
- **Last release - SRB 3.0.1 in December, 2003**
- **Next release - SRB 3.1 in April, 2004**

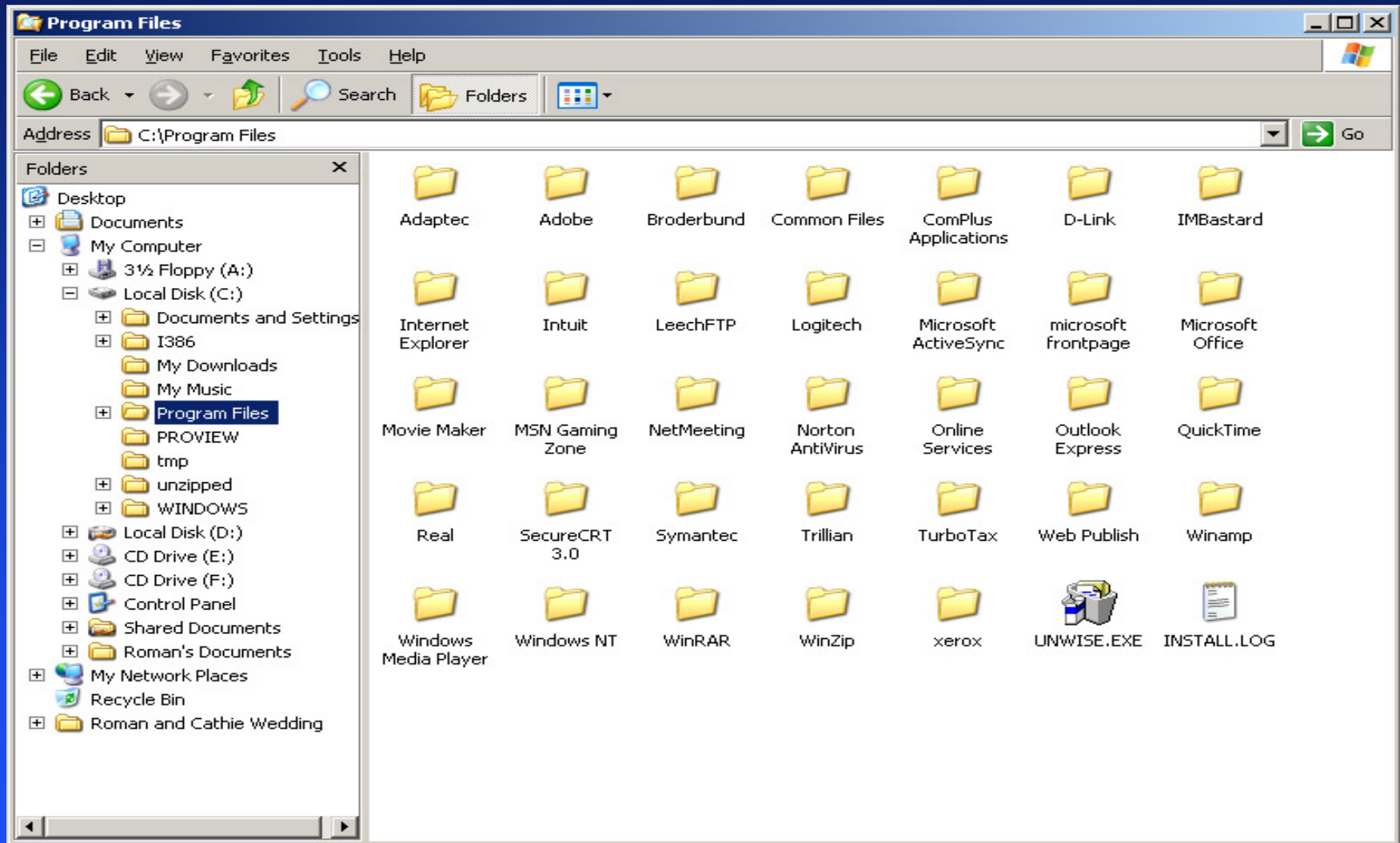
# SRB Problem Reporting

- **srb-chat@sdsc.edu**
  - SRB user community posts problems and solutions
- **srb@sdsc.edu**
  - Request copy of source
- **<http://www.npaci.edu/DICE/SRB/>**
  - Access FAQ, installation instructions, papers

# SRB APIs

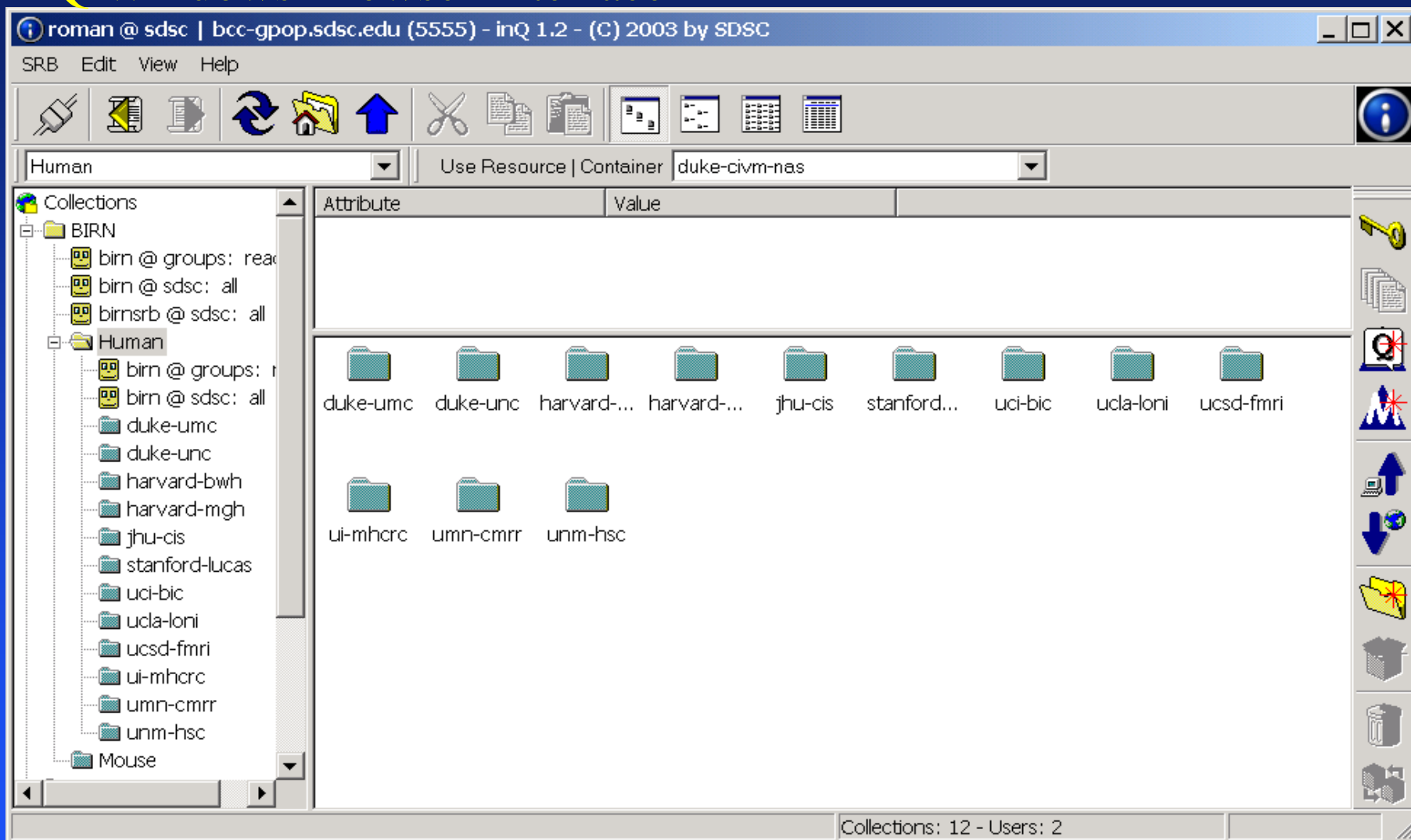
- **C library calls**
  - Provide access to all SRB functions
- **Shell commands**
  - Provide access to all SRB functions
- **mySRB web browser**
  - Provides hierarchical collection view
- **inQ Windows browser**
  - Provides Windows style directory view
- **Jargon Java API**
  - Similar to java.io. API
- **Matrix WSDL/SOAP Interface**
  - Aggregate SRB requests into a SOAP request. Has a Java API and GUI
- **Python, Perl, C++, OAI, Windows DLL, Mac DLL, Linux I/O redirection, GridFTP (soon)**

# What we are familiar with ...



# What we are not familiar with, yet =)

## inQ Windows Browser Interface



# How do they differ?

- Folder, does NOT mean physical folder
- Files, do NOT mean physical files
- Everything is logical
- Everything is distributed
- Permissions are NOT `rw-rw-rw-`
- Permissions are on an object by object basis

# inQ

- **Windows OS only**
- **User Guide at**  
<http://www.npaci.edu/dice/srb/inQ/inQ.html>
- **Download .exe from**  
<http://www.npaci.edu/dice/srb/inQ/downloads.html>



# inQ Features

- **Familiar “Windows Explorer” GUI**
  - Menus
  - Buttons
    - Top: Explorer like
    - Side: Common SRB operations
  - Pull-downs
    - Metadata
    - Resource/container
  - Graphical navigation
    - Plus/minus sign for permissions & subcollections
    - Drag and drop

# inQ Notes

- can store connection parameters
- pay attention to default resource
- upload limited files using up arrow
- upload unlimited files using drag and drop
- download via arrow or drag and drop

# inQ Notes (cont'd)

- **viewing and setting permissions**
  - Recursive?, click now
  - “Add”
  - Domains or Groups?
- **adding metadata**
- **querying metadata, use AND to join small queries into a complex one**

# mySRB

- Web-based access to the SRB
- Secure HTTP
- <https://srb.npaci.edu/mySRB2v7.shtml>
- Uses Cookies for Session Control

# mySRB Features

- Access to Both Data and Metadata
- Data & File Management
- Collection Creation and Management
- Metadata Handling
- Browsing & Querying Interface
- Access Control
- New file creation without upload

# mySRB Interface to a SRB Collection

## View All Metadata

Collection: **Images**  
 Parent Collection: **/home/testuser.sdsc/CDLTest**  
 Owner: **testuser@sdsc**

### Metadata describing entity *Images*



/home/testuser.sdsc/CDLTest/Images

Function	Data Name	Creation Time	Owner	Replica Number	Version Number	Size	Data Type	Resource	In Container
Get File	<a href="#">I0024586A.jpg</a>	2001-07-19-16.08.43	testuser@sdsc	0	0	174713	jpeg image	test-unix	No
Get File	<a href="#">I0024586A.jpg</a>	2001-11-12-11.18.54	testuser@sdsc	1	0	174713	jpeg image	ora-sdsc	No
Get File	<a href="#">I0024586A.jpg</a>	2001-07-20-16.14.05	testuser@sdsc	2	0	174713	jpeg image	hpss-sdsc	No
Get File	<a href="#">I0024586B.jpg</a>	2001-07-20-10.35.55	testuser@sdsc	0	0	44719	jpeg image	test-unix	No
Get File	<a href="#">I0024586C.gif</a>	2001-07-19-16.02.09	testuser@sdsc	0	0	17933	gif image	test-unix	No
Get File	<a href="#">I0024587A.jpg</a>	2001-07-20-11.34.48	testuser@sdsc	0	0	213245	jpeg image	test-unix	No
Get File	<a href="#">I0024587B.jpg</a>	2001-07-20-11.35.33	testuser@sdsc	0	0	59848	jpeg image	test-unix	No
Get File	<a href="#">I0024587C.gif</a>	2001-07-20-11.32.35	testuser@sdsc	0	0	19646	gif image	test-unix	No
Get File	<a href="#">I0024588A.jpg</a>	2001-07-20-11.38.53	testuser@sdsc	0	0	159656	jpeg image	test-unix	No
Get File	<a href="#">I0024588A.jpg</a>	2001-07-27-15.53.35	testuser@sdsc	1	0	159656	jpeg image	hpss-sdsc	No
Get File	<a href="#">I0024588B.jpg</a>	2001-07-20-11.39.14	testuser@sdsc	0	0	42939	jpeg image	test-unix	No
Get File	<a href="#">I0024588C.gif</a>	2001-07-20-11.37.36	testuser@sdsc	0	0	19707	gif image	test-unix	No
Get File	<a href="#">I0024589A.jpg</a>	2001-07-20-11.42.39	testuser@sdsc	0	0	175867	jpeg image	test-unix	No
Get File	<a href="#">I0024589B.jpg</a>	2001-07-20-11.43.02	testuser@sdsc	0	0	47507	jpeg image	test-unix	No
Get File	<a href="#">I0024589C.gif</a>	2001-07-20-11.42.01	testuser@sdsc	0	0	19595	gif image	test-unix	No
Get File	<a href="#">I0024590A.jpg</a>	2001-07-20-11.45.48	testuser@sdsc	0	0	180572	jpeg image	test-unix	No
Get File	<a href="#">I0024590B.jpg</a>	2001-07-20-11.46.13	testuser@sdsc	0	0	40202	jpeg image	test-unix	No



# Provenance Metadata

## View All Metadata

Data Object: **I0024586A.jpg**  
Parent Collection: **/home/testuser.sdsc/CDLTest/Images**  
Owner: **testuser@sdsc**

---

### Metadata describing entity **I0024586A.jpg**

*Field Name : Value Units < Explanation >*

DC.Title : **View Up the Yo Semite [Yosemite] Valley**

DC.Title.Alternate : **Watkins (Carleton E.) Views of San Francisco Yosemite and Monterey; ca. 1876 - ca. 1890**

DC.Creator.Photographer : **Watkins; Carleton E.**

DC.Publisher : **Bancroft Library**

DC.Date : **ca. 1876 - ca. 1890**

DC.Type : **Image**

DC.Format : **image/jpeg**

DC.Identifier : **id:0018654**

DC.Source : [Source Document](#) URL

DC.Language : **ISO639-1; en**

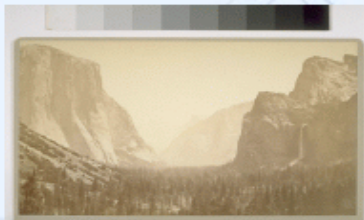
DC.Coverage.PeriodName : **ca. 1876 - ca. 1890**

DC.Coverage.PlaceName : **San Francisco, California**

DC.Rights : [Rights Document](#) URL

Resolution : **hi-res**

color : **beige**



ThumbNail Image :

[srbImage](#)

MediumSize Image : </home/testuser.sdsc/CDLTest/Images/I0024586B.jpg>

[srbObject](#)

# Scommands

- **Command line access to the SRB**
- **Login to machine with Scommand binaries**
- **Verify/Create ~/.srb/.MdasEnv**
- **Verify/Create ~/.srb/.MdasAuth**



# ~/.srb/.MdasEnv file

- **mdasCollectionHome** '/home/kremenek.npaci'
  - Logical path name for collection
- **mdasDomainHome** 'npaci'
- **srbUser** 'kremenek'
  - The combination DomainHome/srbUser defines a user
- **srbHost** 'srb.sdsc.edu'
  - Location of MCAT catalog
- **srbPort** '5615'
  - Port for accessing MCAT catalog
  - The combination srbHost/srbPort defines the catalog

# **.MdasEnv, .MdasAuth**

- **valid authorization schemes are : 'PASSWORD\_AUTH', 'ENCRYPT1', 'GSI\_AUTH'**
  - ENCRYPT1 is a challenge/response mechanism
  - GSI-AUTH is Grid certificate mechanism
- **defaultResource 'dl1-unix-sdsc'**
  - Default location for storage repository
- **File ~/.srb/.MdasAuth contains the SRB password**

# Scommand Features

- **Command line interface**
  - SCRIPTING
  - BATCH and Workflow upload/download
- **Flexibility**
  - Power
  - Complexity
- **Installed man pages via “man [Scommand]”**

# Scommand Features (cont'd)

- **Shelp**
  - Gives list of commands with brief summary
  - “[Scommand] <return>” gives usage info (usually)
- **Sinit** – establishes connection
- **Senv** – displays connection information
- **Sexit** – ends connection

# Navigation Commands

- Spwd
- Senv
- Spasswd
- Serror -3219

# Serror number

- describes SRB errors
- takes an error number generated by SRB/MCAT system and displays a text human readable message relating to the error

# Spasswd

- **changes password of current user**
- **changes the current user's password both in the Meta Catalog as well as in the Client Authorization Environment file**
- **password change persists across sessions with SRB**

# Sexit

- **Sexit**
  - Terminate session
- **Sattr**
  - Lists the queriable MCAT attributes used in conditions for choosing SRB objects



# Simple File Ingestion and Access

- **Example use of commands**
- **cat /tmp/SP2.srb** - list local file
- **Smkdir SP2** - make a SRB collection
- **Sls -l** - list the current SRB collection
- **SgetColl SP2** - display information about collection
- **Sls -l SP2** - list the SRB collection
- **Scd SP2; Spwd** - move to the SP2 collection
- - list the SRB location

# Collection Examples

- **Smv remote\_text\_file remote\_text\_file2**
  - Changes the collection for objects in SRB space
- **SgetD remote\_text\_file2**
  - Display information about SRB data object
- **Srm -pr SP2**
  - Remove file from SRB space
- **Spwd**
- **Sls -l**
- **Smkdir SP2**
- **Sls -l; Srmdir SP2; Sls -l**

# Smkdir {s+l}

- **Smkdir {[-N] [-c container] collection}**
- **creates a new SRB collection in a format [<path\_name>/]<new\_collection\_name>.**
  - Can give either absolute or relative path
- **-N option overrides the inheritance of a container from parent collection**

# Scd [collection], Spwd

- **Scd {[collection]}**
  - changes the working SRB collection
  - without a collection the mdasCollectionHome value in the
  - ~/.srb/.MdasEnv file will become the new working collection.
- **Spwd**
  - displays current working SRB collection

# Sput {s+l}

- **Sput {[-fpravsmMV] [-c container] [-D dataType] [-S resourceName] [-P pathName] [-R retry\_count] [-M] localFileName|localDirectory ... TargetName}**
- **imports one or more local files and/or directories into SRB space**
- **-p prompts, -f force even if object exists, -a force all replicas, -r recursively, -s serial, -m parallel, -M create checksum**
- **Uses server-driven parallel I/O**

# Recursive Put Example

- `Sput -rf /tmp/SRB1 .`
- `Sls -l; Sls -l SRB1`
- `Sls -l SRB1/SRB2`
- `Sls -l SRB1/SRB3`
- `Sls -l SRB1/SRB3/SRB4`
- `Sls -l SRB1/SRB3/SRB5`
- `Scat SRB1/SRB3/SRB4/test4`

# Sget {switches+list}

- **Sget** {[-n n] [-pfrvsmMV] [-A condition] srbObj|Collection ... localFile|localDirectory}
- **exports one or more objects from SRB space into local file system**
  - -n replica number of the object to be copied, -M computes and compares checksum on retrieval -A “<Attr> <CompOp> <Value>” choose srbObj which conforms to the condition, -t specify a ticket for access permission
- **Uses server-driven parallel I/O**

# Recursive Get Example

- **Sget -rf SRB1 .**
- **find SRB1 -print**
- **# cleanup**
- **\rm -r SRB1**
- **Srm -r SRB1**
- **Spwd**
- **Sls -l**
- **Srm '\*emote\_text\_?ile\*'**



# **Sls {s+l}**

- **Sls {[-aChl] [-L number] [-r|-f] [-A condition] [collection|srbObj ...]}**
- **display objects and sub-collections in current SRB working collection or specified SRB collection**
- **-r recursively for sub-collections, -f force each argument to be interpreted as a collection, -l long format ( owner, replica #, physical resource, size, time of creation), -a list metadata**

# Scat {switches + list}

- **Scat {[-C n] [-T ticketFile | -t ticket] [-A condition] srbObj ...}**
- reads each srbObj from SRB to stdout
- -A option, only srbObj which conform to the condition are chosen
- If using a ticket, one need not give a srbObj name

# Store and Retrieve Data Example

- `rm -f local_text_file`
- `date > local_text_file`
- `Sput -vf local_text_file remote_text_file1`
- `Sls -l; Sls -l remote_text_file1`
- `Spwd; Scat remote_text_file1`
- `SgetD remote_text_file1`
- `Sget -vf remote_text_file1 /tmp`

# Sattrs

- lists the queriable MCAT attributes used in conditions for choosing SRB objects.

# Simple Cleanup

- Srm
- Sl
- Srmkdir
- Sl
- Srm -r
- Sl

# Srm {s+l}

- Srm {[-n replicaNum] [-pu] [-A condition] srbObj ...}
- Srm {[-p] [-A condition] -ru srbObj|collection ...}
- remove files from SRB space
- -p prompts, -r recursively (the collection will be emptied of datasets and removed), -u unregister the data from MCAT, the physical file is not removed.

# Srmdir collection

- deletes an existing SRB collection

# System Metadata Discovery

- SgetR
- Stoken
- SgetU
- SgetD
- SgetColl



# SgetU {switches+list}

- {[-pPhdatg] [-L number] [-Y number] [-T userType]  
[userName@domainName]}
- displays information about a group or user  
userName@domainName
- -p user/group name, -a access permissions, -d  
domain(s), -t audit info. -g group(s), -c collection  
access, -T info. for user type

# SgetD {switches+list}

- SgetD {[-phPrReasdDc] [-I] [-W | -U userName] [-Y number] [-L number] [-P dataType] [-A condition] [dataName]}
- display information about SRB data objects
- -p basic parameters, -r storage information, -a permissions, -d audit info., -c collection info., -W for all users, -Y [number] format, -L display number of items at a time

# SgetR {switches+list}

- **SgetR {[-lhdDp] [-L number] [-Y number] [-T resourceType] [resourceName]}**
- **display information about SRB resource(s)**
- **-l display comprehensive list, -d list objects, -D with details, -p for physical resources only, -T resource type list for the given type, -Y [number] controls spacing in display format**

# Data Movement and Data Replication

- Scp
- Smv
- Sreplicate
- Scp -r
- Smc <collection>
- Sphymv
- Sput <logical resource>

# Scp {switches + list}

- **Scp** {**[-n n ] [-fptra] [-c container] [-S newResourceName] [-P newPathName]**
  - srcObj destObj
  - srcObj ... target collection
  - -r source collection... target collection }
- **Copies a srbObj or srbCollection in SRB space**
- **-p prompts, -f force, -a force all replica, -r copy recursively, -n replica number**

# Sreplicate {s+l}

- **Sreplicate {-n replicaNum} [-pr] [-S resourceName] [-P pathName] srbObj|collection ...}**
- **makes one more copy of srbObj or collection**
- **-p prompts, -r recursively, -n replicaNum, -P full or relative newpathName to move the object, -S new resourcename**

# **Smv {s+l}**

- **{srbObj targetObj}**
- **{collection newcollection}**
- **{srbObj ... Collection}**
- **Changes the collection for objects in SRB space**

# Sphymove {s+l}

- Sphymove {[-C n] [-p] [-P newpathName] srbObj ... newresourceName}
- moves one or more SRB objects to the newresourceName at new path newpathName (if given ). The old copy is deleted and the MCAT catalog is also updated



# Replication Examples

- **Sput -vf local\_text\_file remote\_text\_file**
- **SgetD remote\_text\_file**
- **Sreplicate -S "du-sdsc-hpss" remote\_text\_file**
- **SgetD remote\_text\_file**
- **Sreplicate -S "du-caltech-hpss" remote\_text\_file**
- **Sls -l**
- **SgetD remote\_text\_file**
- **Srm -n 0 remote\_text\_file**

# Modifying System Metadata

- **Schmod**
- **SmodD**
- **SmodColl**

# Schmod {switches+list}

- Schmod {[-c] [-a] [-p] [-r] [-d|c] w|a|r|n newUserName domainName collection| srbObj ...}
- grants/changes access permits for the operand collection or srbObj ... for newUserName in domainName
- granted new permission for all replica

# SgetColl {switches+list}

- SgetColl {[-ahc] [-I] [-W | -U userName] [-Y number] [-L number] [-A condition] [collName]{
- display information about SRB data objects
- -a display permissions, -W all users, -c container, -U for usr/group, -I in all collections, -Y output format, -A condition option "<Attr> <CompOp> <Value>"

# SmodD {s+l}

- **SmodD {-s|-t|-c newValue dataName}**
- **modifies metadata information about SRB data objects**
- **-s change size**
- **-t change type**
- **-c insert comments**

# User-defined Metadata

- Sannotate
- Smeta <ingest for data>

# Sannotate {switches}

- {-w position annotation dataName}
- {-u timestamp newAnnotation dataName}
- {[-R] [-t timestamp] [-p position] [-U userName@domainName] [-Y n] [-L n] [-T dataType] dataName|collectionName}
- facility for annotations on data objects

# Smeta {s+l}

- **modifies metadata** information about SRB data objects
- **{-i -I metaAttrNameEqValue [ [-I metaAttrNameEqValue] ... ] dataName}**
- **{-u metadataIndex metaAttrNameEqValue dataName}**
- **{-d metadataIndex dataName}**
- **{-c -i -I metaAttrNameEqValue [ [-I metaAttrNameEqValue] ... ] collectionName}**
- **{-c -u metadataIndex metaAttrNameEqValue collectionName}**



## Smeta cont.

- {-c -d metadataIndex collectionName}
- {[-R] [-I metaAttrNameOrCondition [-I metaAttrNameOrCondition] ... ] [-Y n] [-L n] [-T dataType] dataName|collectionName}
- {-c [-R] [-I metaAttrNameOrCondition [-I metaAttrNameOrCondition] ...] [-Y n] [-L n] [-T dataType] [collectionName]}

## Smeta cont.

- Smeta provides facility for inserting, deleting, updating and accessing meta-data on data object dataName or collection collectionName
- Currently, we support 10 string attributes and two integer attributes
- 'all' permission for modify, 'read' for view

# SmodColl {s+l}

- **SmodColl {[-dh] -c value collName}**
- **modifies information about collections in collName**
- **-h help, -d delete, -c container\_name is updated**

# Smkcont {s+l}

- **Smkcont {[-S resourceName] [-D dataType] [-s containerSize] container}**
- **creates a new SRB container**
- **“container” may be an absolute path or a relative path (will be created in the user's container collection path - /container/userName.domainName)**

# Slscont {s+l}

- Slscont {[-a] [-l]} or {container}
- display metadata of SRB containers
- “Slscont” displays all containers
- “Slscont XYZ” all inContainer objects will be listed
- -l metadata in long format,-a accessible by the user rather than owned by the user

# Srmcont {s+l}

- Srmcont {[-f] container}
- remove an empty existing SRB container
- -f Force the removal of all inContainer objects stored in this container before removing the container

# Sreplcont {s+l}

- Sreplcont {-S resource container}
- replicate a container copy to a specific resource
- For containers that have multiple "permanent" and "cache" copies, this is a way to put a copy of the container on a specific resource

# Ssyncont {s+l}

- Ssyncont {[-d] [-p] container}
- synchronize the "permanent" copies of the container with the "cache" copy.
- when an inContainer object is created or opened for I/O, all I/O are done only to the "cache" copy
- -d delete cache copy, -p to primary only



# Registration and Shadow Objects

## Sregister {s+l}

- {[-p] [-D dataType] [-S size] [-R resourceName]  
RegisteringObjectPath ... TargetName}
- {-c [-p] [-D dataType] [-S size] [-R resourceName]  
RegisteringObjectPath srbObjectName}
- registers one or more files into SRB space

# Stcat {s+l}

- **Stcat** {[-T ticketFile | -t ticket] [-A condition] hostName srbObj ...}
- **display files read from SRB space for a ticketuser**
- **-T option to give a filename containing a ticket, -t option for giving a ticket directly, -A condition "<Attr> <CompOp> <Value>"**

# Sticket {s+l}

- **Sticket { [-F fileName] [-B beginTime] [-E endTime] [-N AccessCount] { -D dataName | -C collName | -R collName } user@domain ... }**
- **issue tickets for SRB objects and collections**
- **-D option for a single data object, -C option for SRB collection, -R option recursively**

## Stls {s+l}

- **Stls {[-v] [-L number] [-Y number] { [-F fileName] | ticket } [-A condition] hostName}**
- **display objects and sub-collections in SRB collection for a given ticket**
- **-v verbose, -F fileName specifies the file in which the ticket is stored**

# Srmticket {s+l}

- **Srmticket { [-F fileName] | ticket }**
- **removes a previously issued ticket. One has to own the ticket to remove it**
- **-F fileName specifies the file in which the ticket is stored**

# SgetT {switches+list}

- **SgetT** {**[-h] [-u] [-v] [-L n] [-Y n] [ -F fileName | -T ticket | -D dataName | -C collection | -U [-c] userName domainName ]**}
- **display information about SRB tickets for a given ticket, dataName or collection**
- **-u ticket-users perspective, -F file for the ticket, -T ticket, -D dataName, -C collection, -U userName**

# Token {s+l}

- **Token {[-L number] [-Y number] typeName}**
- **Displays information about metadata type typeName**
- **typeName can be one of : ResourceType, DataType (default), UserType, Domain, Action, AccessConstraint**

# Remote Proxy Commands

- **Spcommand** {[-h] [-H hostAddr] command}
- **proxy command operation.** Request a remote SRB server to execute arbitrary commands on behalf of client on the “hostAddr” (or “srbHost” in the .MdasEnv). The command/argument string is quoted.
  - Spcommand “hello -xtz”
  - The host location defaults to the host where the client is first connected (srbHost defined in the .MdasEnv file)
- **the proxy commands should be installed in the /usr/local/srb/bin/commands directory**



# Sappend {switches}

- appends a local or a SRB object to an existing SRB object
- { localFileName srbTarget}
  - Append a local file to an existing SRB object
- {-i srbTarget}
  - Appended file is taken from the standard input
- {-s srbObj srbTarget}
  - Append an existing srbObj to another SRB object

# Sgetappend {s+l}

- **Sgetappend {[-C n ] [-p] [-A condition] srbObj ... localFile}**
- **exports object(s) into local file system and appends to localFile**
- **-p prompts before operation, -C replica number, -A condition list ('&' separated) the form "<Attr> <CompOp> <Value>"**

# Sumeta/Sufmeta

- **Sufmeta** {[-f fileName] [-Q meta-data query string] }
- **Option -f is used to bulk insert metadata**
  - Where fileName is a metadata input file and contains the data identifier, meta-data attribute name, value, comments ...
  - Bulk Meta-data Input file format (example)
    - SETMINMETADATANUM=[GIVENPERDATA | 0 | GETFROMMCAT] /\*\*  
first line \*\*//
    - /home/collection-identifier |dataName|attributeName|value ...(other lines)
- **Option -Q is used to query the MCAT metadata**
  - Can be used to discover data based on the attributed
  - English-like and SQL query constructs supported
  - Examples:
    - Sufmeta -Q brightness between “1000” and “21000”
    - Sufmeta -Q color like “green”

# SRB Installation

- **Installation procedure written by Michael Doherty**
  - SRB\_Install\_Notes.doc
- **Perl install script for Mac OS X and Linux written by Wayne Schroeder**
  - <http://www.npaci.edu/DICE/SRB/tarfiles/install.pl>
  - Installs PostgreSQL, MCAT, SRB server, SRB clients
  - Installation takes 18 minutes on a Mac G4

# For More Information

Reagan W. Moore  
Arun S. Jagatheesan  
San Diego Supercomputer Center

[moore@sdsc.edu](mailto:moore@sdsc.edu)

[arun@sdsc.edu](mailto:arun@sdsc.edu)

<http://www.npaci.edu/DICE/>

<http://www.npaci.edu/DICE/SRB>

<http://www.npaci.edu/DICE/SRB/matrix/>